Path Generation Tactics for a UAV Following a Moving Target

Craig R. Husby

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science in Aeronautics and Astronautics

University of Washington

2005

Program Authorized to Offer Degree: Aeronautics and Astronautics

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
31.May.05

3. REPORT TYPE AND DATES COVERED
THESIS

4. TITLE AND SUBTITLE
PATH GENERATION TACTICS FOR A UAV FOLLOWING A MOVING TARGET

5. FUNDING NUMBERS

6. AUTHOR(S)
2D LT HUSBY CRAIG R

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSITY OF WASHINGTON

8. PERFORMING ORGANIZATION REPORT NUMBER
CI04-1096

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS

15. NUMBER OF PAGES
98

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

18. SECURITY CLASSIFICATION OF THIS PAGE

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

University of Washington

Abstract

Path Generation Tactics for a UAV Following a Moving Target

by Craig R. Husby

Chair of Supervisory Committee:

Professor Rolf Rysdyk
Department of Aeronautics and Astronautics

There is a need for generating paths for a fixed wing aircraft maneuvering to follow a target moving at various speeds. In this thesis we investigate three possible patterns and associated algorithms for an unmanned aerial vehicle (UAV) autonomously following a moving target. The three patterns are a square wave pattern, a circular pattern, and a bow tie-shaped standoff pattern. These patterns range in capability of being able to follow a target that is standing still up to one that is traveling at near UAV airspeed, the best solution being some combination of patterns based on the ratio of UAV velocity to target velocity. The path algorithms assume that the position and velocity is known for the target and that the UAV is already in visual range (by an onboard camera) of the target. Simulation results are shown to confirm the capabilities of each method.

University of Washington

Graduate School

This is to certify that I have examined this copy of a master's thesis by

Craig R. Husby

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Rolf Rysdyk

Juris Vagners

Date: 03-15-'05

In presenting this thesis in partial fulfillment of the requirements for a master's degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Any other reproduction for any purpose or by any means shall not be allowed without my written permission.

Signature Craig R

Date 17 Mar 05

University of Washington

Abstract

Path Generation Tactics for a UAV Following a Moving Target

by Craig R. Husby

Chair of Supervisory Committee:

Professor Rolf Rysdyk
Department of Aeronautics and Astronautics

There is a need for generating paths for a fixed wing aircraft maneuvering to follow a target moving at various speeds. In this thesis we investigate three possible patterns and associated algorithms for an unmanned aerial vehicle (UAV) autonomously following a moving target. The three patterns are a square wave pattern, a circular pattern, and a bow tie-shaped standoff pattern. These patterns range in capability of being able to follow a target that is standing still up to one that is traveling at near UAV airspeed, the best solution being some combination of patterns based on the ratio of UAV velocity to target velocity. The path algorithms assume that the position and velocity is known for the target and that the UAV is already in visual range (by an onboard camera) of the target. Simulation results are shown to confirm the capabilities of each method.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

NOMENCLATURE

Symbols

D_d	desired distance from target
$\mathcal{F}$	reference frame
$\mathcal{F}_s$	Serret-Frenet frame, see Figure 2.2
$\mathcal{F}_e$	Earth frame
g	gravity constant
n	number of segments
r	radius
$r_{(\cdot)}$	radius of segment $(\cdot)$
s	arclength position along desired path
$s_{(\cdot)}$	segment n or length of segment $(\cdot)$
S_T	distance from UAV to target
T_p	time it takes to complete one cycle of the pattern
v	velocity component along the body-fixed y-axis
V	velocity
V_a	airspeed
V_c	groundspeed
V_t	target velocity
W	wind speed
x	position along x-axis
$\mathbf{x_s}, \mathbf{y_s}$	Serret-Frenet axes system
y	position along y-axis
y_s	Normal distance, course deviation, or cross-track error

z^{-1}	one time step delay
α	angle associated with geometry of moving circular pattern (see Figure 4.2)
β	angle associated with geometry of moving circular pattern (see Figure 4.2)
Γ	distance from UAV to target (see Figure 4.2)
γ	angle associated with geometry of moving circular pattern (see Figure 4.2)
θ	bearing from target to UAV (see Figure 4.2)
κ	curvature of path
ν	pseudo-control
σ	sigmoidal squashing function, which forms the nonlinear helmsman tactics
τ	first order time constant
ϕ	bank angle
χ	course
ψ	heading
Ω	angular velocity magnitude

Subscripts

$1, 2, 3...$	segment number 1,2,3...
c	inertial or command
f	feedback
$icpt$	intercept
n	new
s	related to the Serret-Frenet reference frame
t	target
w	wind or due to wind
x	x direction
y	y direction

Superscripts

$\tilde{\cdot}$	error relative to desired path
$\vec{\cdot}$	directed vector quantity

ACKNOWLEDGMENTS

I would like to express sincere appreciation to my research advisor, Dr. Rolf Rysdyk, who diligently guided and encouraged me through this project. I would also like to acknowledge the Henry L. Gray Fellowship for the grant which allowed me to attend the University of Washington and make this project possible. Finally I would like to thank the Air Force for allowing me the time to get this degree.

The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

Chapter 1

INTRODUCTION AND MOTIVATION

Unmanned Aerial Vehicles (UAVs) represent the capability of performing numerous essential operations at both lower cost and lower risk than those of manned aircraft. Currently many UAV operations, such as target following, are conducted manually. One desired advancement in the continual development of UAVs is reaching the point where the UAV is capable of autonomously following a moving target. There are many steps that need to be accomplished to achieve such an advancement. For instance, target recognition is needed. The UAV's computer needs to be able to pick out the object it is following from the images it is being fed by the on board camera. Once that is achieved, it is necessary to have a method of determining the target's position and velocity (or relative position and velocity with respect to the UAV) based solely on those two-dimensional images. Then the UAV needs to be programmed to take that data and fly such that the target remains in view at all times. In other words, a trajectory must be determined that allows the onboard sensor (such as a nose-mounted camera) to keep the target within its range.

A lot of work has been done on autonomously controlling UAVs using waypoints. For this problem however, that is not necessarily the best solution because the future movements of the target are unknown. A better solution would be one that has an initial path that changes frequently as the position of the target is updated.

The focus of this paper is, having real-time data of the targets position and velocity and accurately knowing the UAV's position and velocity (presumably from GPS), developing the code necessary for the UAV to follow the target such that the target remains in constant view of a nose-mounted camera with limited motion. We looked at three possible flight patterns for the UAV: a square wave, a moving circle, and a moving bow tie-shaped pattern.

Simulation results are presented to demonstrate the effectiveness of each pattern.

In order to focus on controlling the flight path, we use a preexisting platform: the Simulink model of the Aerosonde UAV, and a preexisting path-following algorithm: the Serret-Frenet method [16]. The tactical algorithms were designed to integrate with the existing task-driven path generation algorithms and make use of proprietary inner loops that expect a turn rate command. We assume that some other path-planning algorithm already placed the UAV near the target (for purposes of simulation we start the UAV and the target at the same point although this is not necessary). There also exists the possibility with each of these patterns to add some offset such that the UAV remains some arbitrary distance away from the target (i.e. a quarter mile ahead of or behind the target). Another simplification we make is that the UAV has a constant airspeed of $25m/s$.

1.1 Related Work

There are several other institutions conducting research in this area of interest. In regards to following a target, Girard, et. al., in research supported by the Office of Naval Research, tackled the problem by using a circular pattern for low speeds and then transitioning into a sinusoidal pattern for higher speeds [7]. We looked at the sinusoidal pattern, created at the University of California, Berkeley [10], in which they vary the amplitude of the sinusoid in order to adjust to the targets speed. With this method they were able to follow a target moving at the same speed as the UAV down to a ratio of the UAV's speed to the target's speed of 3:1. For reasons of integration, and for operation in windy environments, we feel that this is not the most efficient method of accomplishing the task because it relies on an arbitrary period T that determines the path and does not take into account the capability of the UAV to fly the turns necessary to follow the path generated.

Other work is being done on how a networked team of autonomous vehicles can accomplish more complex tasks or the same tasks more efficiently [15].

Research is also being conducted on tracking the target using video images. Currently this technology can locate targets from images using algorithms that recognize pixel patterns [12]. Once the target is identified in one frame, a tracking algorithm estimates the target trajectory by looking at multiple frames over time. Dr. Boyd, with the Acquisition, Querying and Prediction of Motion Trajectories project, funded by IRIS Precarn (Canada), is looking at trajectory acquisition from video, storage and retrieval of trajectories from a database, and predicting the future path of an object based on previously observed trajectories [8]. Other work in this area includes particle filters for moving cameras without stabilization [9] and the Joint Probabilistic Data Association Filter (JPDAF) for tracking multiple targets with unreliable target identification [6].

1.2 Applications

The US Navy, US Marine Corps, US Army, UK Royal Air Force, UK Navy, and Japanese Coast Guard have all indicated to us that they are interested in the capability of autonomously tracking targets of interest [8]. According to the Department of Defense, unmanned vehicles, including UAVs, will be a part of its future combat systems [4]. On the civilian side, the US as well as some foreign governments have shown interest in this technology for non-military applications such as tracking fish, marine mammals, and lifeboats for search and rescue missions [8]. Tracking ocean debris such as fishing nets in the Northeast Pacific Ocean would greatly benefit the environment since they are a significant threat to wildlife [5].

Chapter 2

UAV PLATFORM AND PATH FOLLOWING STRATEGIES

The platform on which we tested our path designs is the Aerosonde UAV (see Figure 2.1), in Matlab's Simulink environment (see Appendices).

In order to implement our design it is necessary to have a path following system. We used the Serret-Frenet method [16]. This requires the input of κ, V_c, y_s, χ_m, and χ_s. Not only did we use this for purposes of following the path we generated, in the case of the circular method and part of the standoff method, we based the path on this technique. The purpose of designing the system in this manner was to allow for its integration with a path planning system, specifically the Evolution-Based Cooperative Planning System (ECoPS) [14].

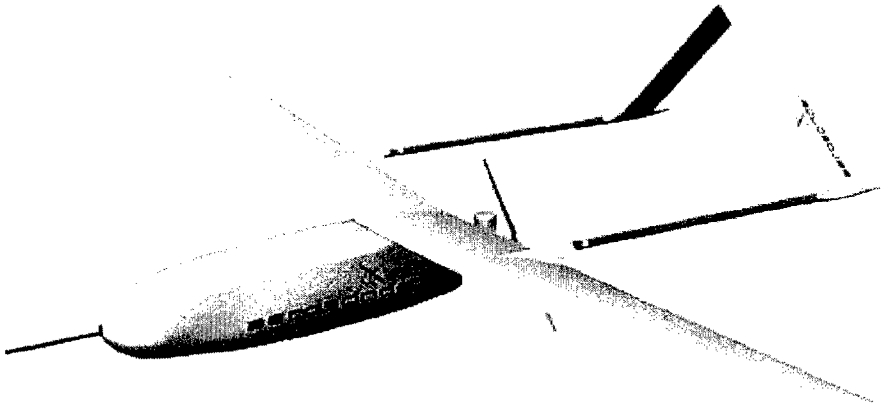


Figure 2.1: This is the Aerosonde UAV which is the aircraft on whose Simulink model we designed and tested our flight patterns [1].

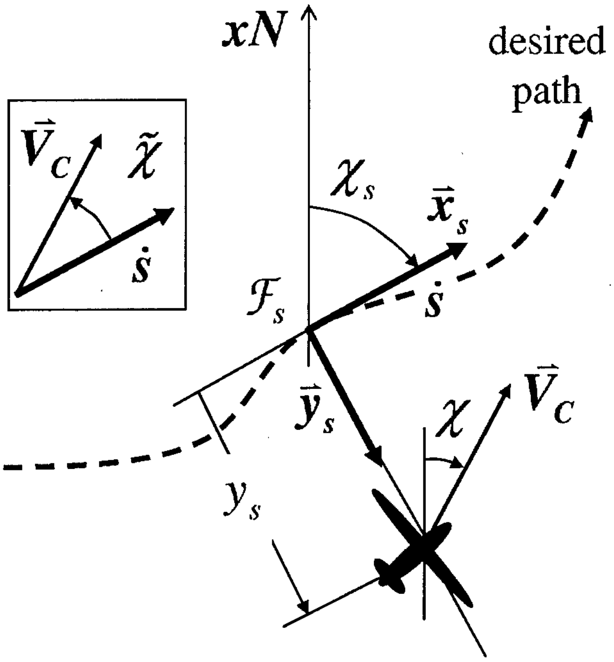


Figure 2.2: The 'Serret-Frenet" frame for a 2D-Path. The Serret-Frenet frame provides a means to ride along the 2D curve and illustrate its properties (curvature) [16].

2.1 Serret-Frenet Method

Consider a frame along the desired path, $\mathcal{F}_s$, with $\mathbf{x_s}$, the unit vector, in the direction of the desired inertial velocity, i.e. tangential to the path, and let $\mathbf{y_s}$ be the unit normal to the path (see Figure 2.2). If the direction of the path is indicated as χ_s (which is considered the desired 'course' when on the path) then

$$\vec{\mathbf{x}}_s = \begin{bmatrix} cos(\chi_s) \\ sin(\chi_s) \end{bmatrix}, \text{ and } \vec{\mathbf{y}}_s = \begin{bmatrix} -sin(\chi_s) \\ cos(\chi_s) \end{bmatrix}. \tag{2.1}$$

At any point along the desired path, its curvature, $\kappa(s)$, is defined as $\kappa(s) = 1/r(s)$, where $r(s)$ is the radius of the path at that point. The Serret-Frenet formulas in 2D provide the following relations.

$$\frac{d\mathbf{x_s}}{ds} = \kappa\mathbf{y_s} \tag{2.2}$$

$$\frac{d\mathbf{y_s}}{ds} = -\kappa\mathbf{x_s} \tag{2.3}$$

$$\dot{\chi}_s \;=\; \kappa \dot{s} \tag{2.4}$$

Therefore, consider a frame, $\mathcal{F}_1$, which is rotated relative to the inertial frame, $\mathcal{F}_e$, over the course angle χ.

$$\begin{aligned} \vec{\mathbf{v}}_1|_s \;=\; {}_sT_1 \begin{bmatrix} u \\ v \end{bmatrix}_1 \;=\; {}_sT_1 \begin{bmatrix} V_C \\ 0 \end{bmatrix}_1 &\;=\; {}_e\vec{\mathbf{v}}_1|_s + ({}_1\vec{\omega}_s \times {}_s\vec{\mathbf{p}}_1)|_s & (2.5) \\ &\;=\; {}_e\vec{\mathbf{v}}_s|_s + {}_s\vec{\mathbf{v}}_1|_s + ({}_1\vec{\omega}_s \times {}_s\vec{\mathbf{p}}_1)|_s & (2.6) \end{aligned}$$

Let the *relative course* be defined as

$$\widetilde{\chi} \triangleq \chi - \chi_s \tag{2.7}$$

then

$$\begin{pmatrix} cos(\widetilde{\chi}) & -sin(\widetilde{\chi}) \\ sin(\widetilde{\chi}) & cos(\widetilde{\chi}) \end{pmatrix} \begin{pmatrix} V_C \\ 0 \end{pmatrix} = \begin{pmatrix} \dot{s} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \dot{y}_s \end{pmatrix} + \begin{pmatrix} 0 & -\dot{\chi}_s \\ \dot{\chi}_s & 0 \end{pmatrix} \begin{pmatrix} 0 \\ y_s \end{pmatrix} . (2.8)$$

This, combined with the fact that $\dot{\widetilde{\chi}} = \dot{\chi} - \dot{\chi}_s$, may be rearranged into

$$\begin{pmatrix} 1 - \kappa y_s & 0 & 0 \\ 0 & 1 & 0 \\ \kappa & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{s} \\ \dot{y}_s \\ \dot{\widetilde{\chi}} \end{pmatrix} = \begin{pmatrix} cos(\widetilde{\chi}) & -sin(\widetilde{\chi}) & 0 \\ sin(\widetilde{\chi}) & cos(\widetilde{\chi}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_C \\ 0 \\ \dot{\chi} \end{pmatrix} . \tag{2.9}$$

Limitation: Equation 2.9 is not usable when $y_s = \frac{1}{\kappa(s)} = r(s)$, i.e. when the vehicle is at the center of the instantaneous circle.

2.2 Helmsman Behavior

The guidance law objective is to converge $\widetilde{\chi}$ and y_s simultaneously to zero. This may be achieved by coupling the commanded angle of convergence and cross-distance, i.e. $\widetilde{\chi}_c(y_s)$, (see Figure 2.3). According to Pettersen and Lefeber [13], the behavior of a 'good helmsman' follows an intercept course $\widetilde{\chi}_c$ that varies with cross distance y_s, rather than using sideward velocity. This is similar to an aircraft in coordinated flight, $v_b \equiv 0$, while considering bank angle as the control variable.

The helmsman behavior can be modeled as $\tilde{\chi}_c = \chi_c - \sigma(y_s)$, where χ_c represents the commanded absolute course and $\sigma(y_s)$ is any function satisfying

$$\begin{aligned} y_s\sigma(y_s) &> 0 \\ \sigma(0) &= 0 \\ \sigma : y_s &\rightarrow [-\tilde{\chi}_{icpt}, \tilde{\chi}_{icpt}] \end{aligned}$$

where $\tilde{\chi}_{icpt}$ represents the intercept angle at large cross distance. The helmsman behavior relative to a straight line course $\chi_s = 0$ is

$$\chi_c = \sigma(y_s). \tag{2.10}$$

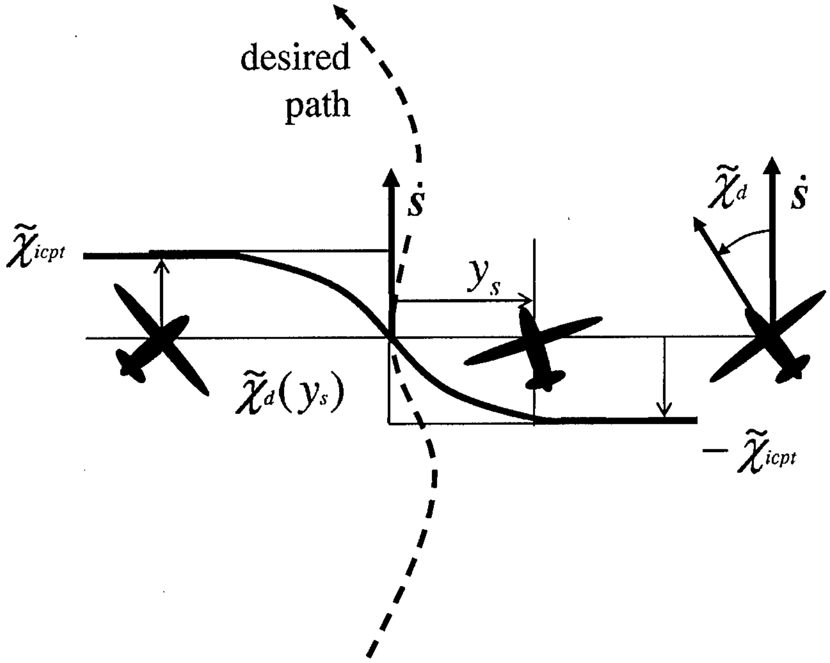


Figure 2.3: The behavior of a 'good helmsman' modeled by the desired relative course $\tilde{\chi}_d$ as function of cross-distance y_s in the form of a sigmoid function (solid line), saturating at $\pm\tilde{\chi}_{icpt}$. $\tilde{\chi}_c \approx 0$ when 'close' to the trajectory, and saturates at $\tilde{\chi}_c = \pm\tilde{\chi}_{icpt}$ when farther away.[16].

In addition distinguish:

- the flown course χ,
- the trajectory direction χ_s,
- the *relative course* $\widetilde{\chi} \triangleq \chi - \chi_s$, and
- the *commanded relative course* $\widetilde{\chi}_c \triangleq \chi_c - \chi_s$.

For intercepting and tracking a trajectory, the helmsman behavior is relative to the desired trajectory. The helmsman behavior is expressed by formulating the commanded course χ_c in terms of χ_s and y_s. In that case, Equation 2.10 becomes

$$\widetilde{\chi}_c \triangleq \chi_c - \chi_s = \sigma(y_s). \tag{2.11}$$

Therefore, the helmsman behavior in trajectory tracking is

$$\chi_c(y_s, \chi_s) = \sigma(y_s) + \chi_s. \tag{2.12}$$

2.3 Convergence Properties

Two aspects of the helmsman behavior determine the 'aggressiveness' of the intercept: the maximum intercept angle and the 'lead-distance' or slope $d\sigma/dy_s$.

Assumption 1 Wind $\{W, \chi_w\}$ are assumed known (estimated).

Assumption 2 The airspeed V_a and altitude remain approximately constant.

Assumption 3 Bank angle command following performs well and fast relative to path-changes of $\pm 30^o$.

Assumption 4 The commanded path will be mild enough to prevent extreme wind-up of path-following integrator due to roll-rate and saturation limits. This is a temporary assumption since some form of hedging of the commanded heading rate can be implemented later.

Let the ideal course convergence dynamics be specified as follows:

$$\frac{d}{dt}\chi(t) = \nu(\chi, \chi_c) \tag{2.13}$$

where 'pseudo-control' ν refers to the 'tracking-servo' control law (similar to Ω_c in [11]). Defined in terms relative to the trajectory we would have

$$\begin{aligned} \frac{d}{dt}\{\chi - \chi_s\} &= \nu(\chi - \chi_s, \chi_c - \chi_s) \\ \frac{d}{dt}\tilde{\chi} &= \nu(\tilde{\chi}, \tilde{\chi}_c). \end{aligned} \tag{2.14}$$

The guidance law then is based on desired tracking dynamics by design of the pseudo control ν. An example is the tracking servo given in Reference [11]. Expressed in the above trajectory parameters,

$$\begin{aligned} \nu &= \Omega_c && (2.15) \\ &= \dot{\tilde{\chi}}_c(y_s, V_C, \tilde{\chi}) + \frac{1}{\tau_i}\int_0^t \left(\dot{\tilde{\chi}}_c - \dot{\tilde{\chi}}\right) \cos(\tilde{\chi}_c - \tilde{\chi})\ d\tau + \tau_{ff}\frac{d}{dt}\dot{\tilde{\chi}}_c. && (2.16) \end{aligned}$$

To ensure smooth and fast convergence in Reference [11], the signal $\dot{\tilde{\chi}}_c(y_s, V_C, \tilde{\chi})$ is determined iteratively, based on relative location, orientation, and aircraft bank performance. Therefore, the helmsman behavior in Reference [11] is defined in terms of course rate-of-change and is a more sophisticated version of Equation 2.12. Comparing Equations 2.11 and 2.12 with the definition in Reference [11]:

$$\begin{aligned} \dot{\chi}_c(y_s, V_c, \tilde{\chi}) &= \Omega^*(y_s, v_s) + \dot{\chi}_s && (2.17) \\ &= \dot{\tilde{\chi}}_c(y_s, v_s) + \dot{\chi}_s && (2.18) \\ &= \dot{\tilde{\chi}}_c(y_s, V_C, \tilde{\chi}) + \dot{\chi}_s. && (2.19) \end{aligned}$$

In this work we consider the design:

$$\nu(\chi, \chi_c) \triangleq \dot{\chi}_c + k_p(\chi_c - \chi) + k_i\int_0^t (\chi_c - \chi)\, d\tau \tag{2.20}$$

in terms of the course relative to the desired path Equation 2.14. This becomes

$$\begin{aligned} \nu(\chi - \chi_s, \chi_c - \chi_s) &= (\dot{\chi}_c - \dot{\chi}_s) + k_p(\chi_c - \chi_s - \chi + \chi_s) + k_i\int_0^t (\chi_c - \chi_s - \chi + \chi_s)\, d\tau \\ \Leftrightarrow\ \nu(\tilde{\chi}, \tilde{\chi}_c) &= \dot{\tilde{\chi}}_c + k_p(\tilde{\chi}_c - \tilde{\chi}) + k_i\int_0^t (\tilde{\chi}_c - \tilde{\chi})\, d\tau. \end{aligned} \tag{2.21}$$

Herein $\widetilde{\chi}_c = \chi_c - \chi_s$ denotes the commanded relative course based on the helmsman behavior, Equation 2.12, displayed in Figure 2.3 and defined by

$$\sigma(y_s) \;=\; \widetilde{\chi}_{icpt} \frac{e^{-a\,y_s/2} - 1}{e^{-a\,y_s/2} + 1}, \tag{2.22}$$

where $a > 0$ is a design variable such that $a\,\widetilde{\chi}_{icpt}$ represents the slope $d\sigma/dy_s$ at $y_s = 0$ (compare to a derivative gain). Therefore, the derivative signal $\dot{\chi}_c$ is constructed as

$$\begin{aligned} \dot{\chi}_c &= \frac{d}{dt}\sigma(y_s) && (2.23)\\ &= \sigma_{y_s}\frac{dy_s}{dt} && (2.24)\\ &= \sigma_{y_s} V_C\, sin(\widetilde{\chi}) && (2.25) \end{aligned}$$

where

$$\sigma_{y_s} \triangleq \frac{d}{dy_s}\sigma(y_s) \;=\; -a\,\widetilde{\chi}_{icpt}\,\frac{e^{-ay_s/2}}{(e^{-ay_s/2} + 1)^2}. \tag{2.26}$$

The closed loop dynamics according to Equation 2.14 then are

$$\dot{\widetilde{\chi}} \;=\; \sigma_{y_s} V_C\, sin(\widetilde{\chi}) + k_p(\widetilde{\chi}_c - \widetilde{\chi}) + k_i \int_0^t (\widetilde{\chi}_c - \widetilde{\chi})\, d\tau. \tag{2.27}$$

Chapter 3

SQUARE WAVE PATTERN

Whereas the circular pattern is complex in its implementation, the square wave pattern was relatively straight forward. This pattern was inspired by the sinusoidal method developed by the University of California, Berkeley [10]. We wanted to alter the sinusoidal pattern with the aim of creating feasible trajectories over a larger speed envelope and in significant wind by ensuring that the UAV would always be able to fly the corners defined by the path. Accomplishing this resulted in the square wave pattern.

3.1 Basic Pattern

The pattern (Figure 3.1) consists of four straight segments and four quarter circles, defined in Table 3.1. Once the UAV has flown this pattern, it resets (see Figure 3.2) and repeats the pattern. Each time the pattern resets, the nominal lengths of segments a and b are set

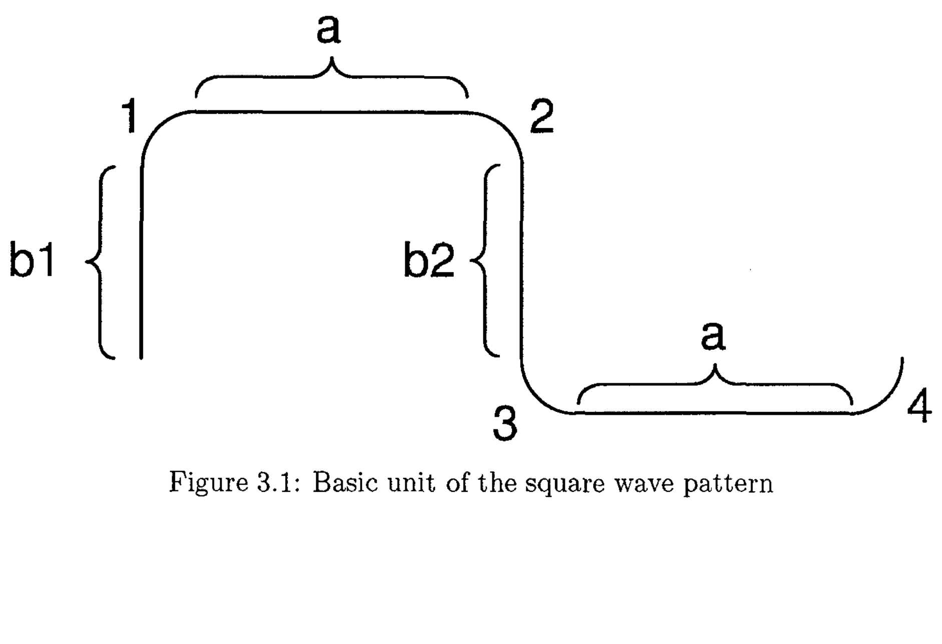


Figure 3.1: Basic unit of the square wave pattern

Table 3.1: Segment definitions for the square wave pattern

Segment	Description
s_{b1}	straight segment with length defined by velocity ratio
s_1	quarter circle with radius determined by turning capability (equal to s_4)
s_a	one of two straight segments of equal length defined by velocity ratio and feedback
s_2	quarter circle with radius determined by turning capability (equal to s_3)
s_{b2}	straight segment with length defined by velocity ratio and feedback
s_3	quarter circle with radius determined by turning capability (equal to s_2)
s_a	one of two straight segments of equal length defined by velocity ratio and feedback
s_4	quarter circle with radius determined by turning capability (equal to s_1)

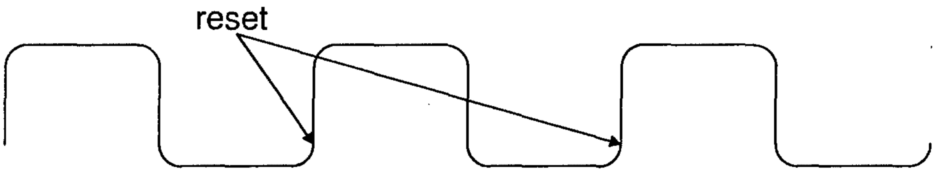


Figure 3.2: Location of square wave reset

based on the UAV's airspeed and the speed of the target according to Equations 3.1 and 3.2.

$$a = \begin{cases} 1 & \frac{V_a}{V_t} - 1 \geq 500 \\ \frac{500}{\frac{V_a}{V_t} - 1} & \frac{V_a}{V_t} - 1 \geq 1 \\ 500 & \frac{V_a}{V_t} - 1 < 1 \end{cases} \tag{3.1}$$

$$b = \begin{cases} 500 & \frac{V_a}{V_t} - 1 \geq 1 \\ 500\left(\frac{V_a}{V_t} - 1\right) & \frac{V_a}{V_t} - 1 < 1 \end{cases} \tag{3.2}$$

These equations cause segment a to increase in magnitude up to a maximum of 500 (arbitrarily chosen) as target velocity increases and segment b to remain constant at 500 until

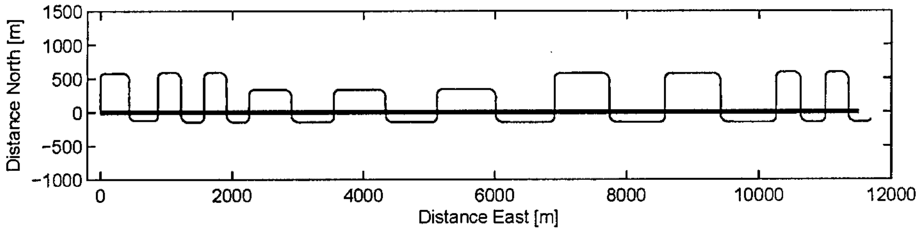


Figure 3.3: UAV's flight path when using the square wave pattern to follow a target with varying speed

that point, when it starts decreasing. Figure 3.3 shows the path resulting from a target changing speed. Target speed is used rather than velocity since a coordinate transformation ensures that the a-segments are always parallel to the target's velocity vector (see Section 3.3). Theoretically, this method would allow the UAV to follow a target moving with a velocity anywhere from a stand still up to the velocity of the UAV by collapsing to a nearly perpendicular track or stretching to a straight parallel course. However, since the UAV cannot make an instantaneous 90° or 180° turn, this is not possible.

The corners are defined by the curvature κ possible, defined by Equation 3.4. The maximum bank angle used is $\phi = 30°$. This is less than the 45° bank angle the UAV is capable of in order to better ensure the curvature is flyable. This robustness is necessary since the velocity used for these calculations are estimated over the whole segment as the ground speed 45° into the turn (Equations 3.5 and 3.6).

$$r = \frac{1}{\kappa} \tag{3.3}$$

$$\kappa = \frac{g \tan \phi}{V_c^2} \tag{3.4}$$

$$V_{c_{1,4}} = V_a + |W| \cos\left(\arctan \frac{W_y}{W_x} - \frac{\pi}{4}\right) \tag{3.5}$$

$$V_{c_{2,3}} = V_a + |W| \cos\left(\arctan \frac{W_y}{W_x} - \frac{5\pi}{4}\right) \tag{3.6}$$

Because the radius of the corners determines the dimensions of the pattern and because the wind determines the radius of the corners, the target speed that the UAV is capable of

Table 3.2: Range of vehicle velocity (m/s) moving Eastward, over which the UAV is capable of tracking the target given an airspeed of $25m/s$ and a wind speed of $10m/s$

	Wind Direction							
	0°	45°	90°	135°	180°	225°	270°	315°
Min	3.24	3.58	4.36	5.89	8.07	8.40	6.20	4.00
Max	19.12	15.52	13.72	15.94	20.96	25.00	25.22	22.59

Table 3.3: Range of UAV velocity to vehicle velocity ratios over which the UAV is capable of tracking the target in a $10m/s$ wind

	Wind Direction							
	0°	45°	90°	135°	180°	225°	270°	315°
Min	7.72:1	6.98:1	5.73:1	4.24:1	3.10:1	3.00:1	4.03:1	6.26:1
Max	1.31:1	1.61:1	1.82:1	1.57:1	1.19:1	1.00:1	0.99:1	1.11:1

following is determined by the wind. Equations 3.7 and 3.8 calculate what the limits are.

$$V_{t_{min}} = \frac{2r_{1,4} + r_{2,3}}{\frac{500}{V_a+W_y} + \frac{\pi r_{1,4}}{V_{c_{1,4}}} + \frac{\pi r_{2,3}}{V_{c_{2,3}}} + \frac{500}{V_a-W_y}} \tag{3.7}$$

$$V_{t_{max}} = \frac{2r_{1,4} + r_{2,3} + 1000}{\frac{1000}{V_a+W_x} + \frac{\pi r_{1,4}}{V_{c_{1,4}}} + \frac{\pi r_{2,3}}{V_{c_{2,3}}}} \tag{3.8}$$

where $W_{x,y}$ are the wind components, the subscripts $1,4$ and $2,3$ refer to the different corners of the pattern defined in Figure 3.1, and $r_{1,4or2,3}$ are given by Equations 3.3 through 3.6. With no wind and a UAV speed of $25m/s$, the range is $4.89m/s$ to $19.65m/s$ or ratios 5.1:1 to 1.3:1. Table 3.2 shows the ranges of target speed possible for different wind directions using a wind speed of $10m/s$ and an Eastward moving target. Table 3.3 shows the same results in terms of velocity ratios.

3.2 Feedback

Also, each time the pattern resets, feedback is used to compensate for any errors in relative location. The difference in **x** and **y** directions are fed back into a and $b1$ according to Equations 3.9 and 3.10. This feedback also accounts for any coordinate transformations due to change in target direction.

$$a_f = \frac{1}{2}\left[\cos(-\Delta\chi_t)N_{k-1} - \sin(-\Delta\chi_t)M_{k-1} + N_k\right] \tag{3.9}$$

$$b_f = -\left[\cos(-\Delta\chi_t)M_{k-1} + \sin(-\Delta\chi_t)N_{k-1} + M_k\right] \tag{3.10}$$

where

$$\Delta\chi_t = \chi_{t_k} - \chi_{t_{k-1}} \tag{3.11}$$

$$M = \int_{t_o}^{t} \left[-\dot{x}\sin(-\chi_t) + \dot{y}\cos(-\chi_t)\right] dt \tag{3.12}$$

$$N = T_x - \int_{t_o}^{t} \left[\dot{x}\cos(-\chi_t) + \dot{y}\sin(-\chi_t)\right] dt \tag{3.13}$$

and

subscript k refers to the segment (where each segment is defined such that χ_t is constant)

subscript o refers to the start of the current segment

T_x is the position of the target in the local x-direction

χ_t is the target heading −90°

3.3 Change in Target's Course

Each time the target changes direction, the angle of change of the course is added to the heading χ_s of the UAV and a coordinate transformation is done such that the feedback (Equations 3.9 to 3.13) is calculated based on the new coordinates as seen in Figure 3.4. Figures 3.5 and 3.6 show this technique in use.

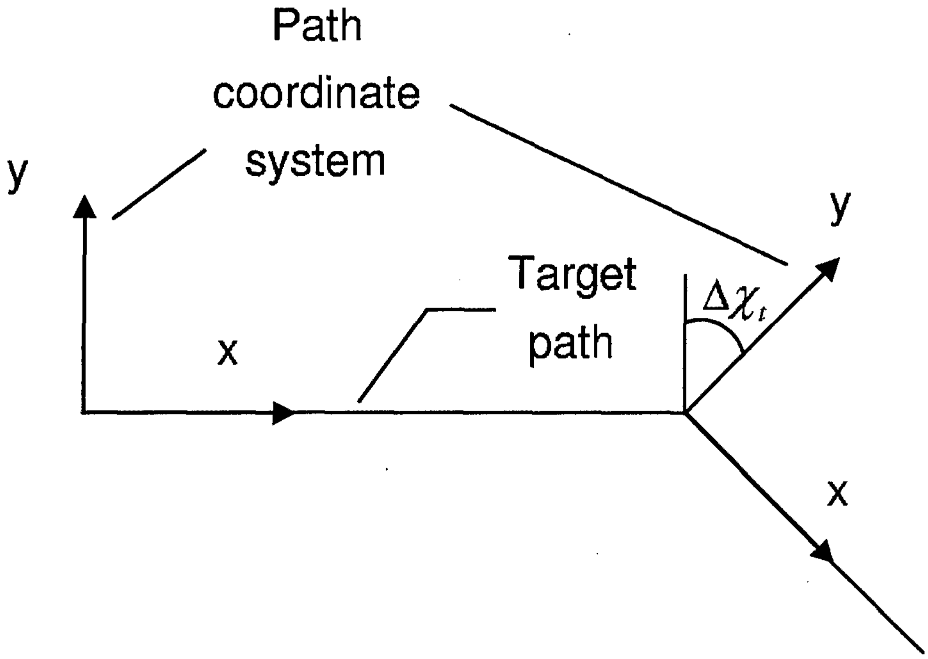


Figure 3.4: Coordinate transformation used when the target changes direction

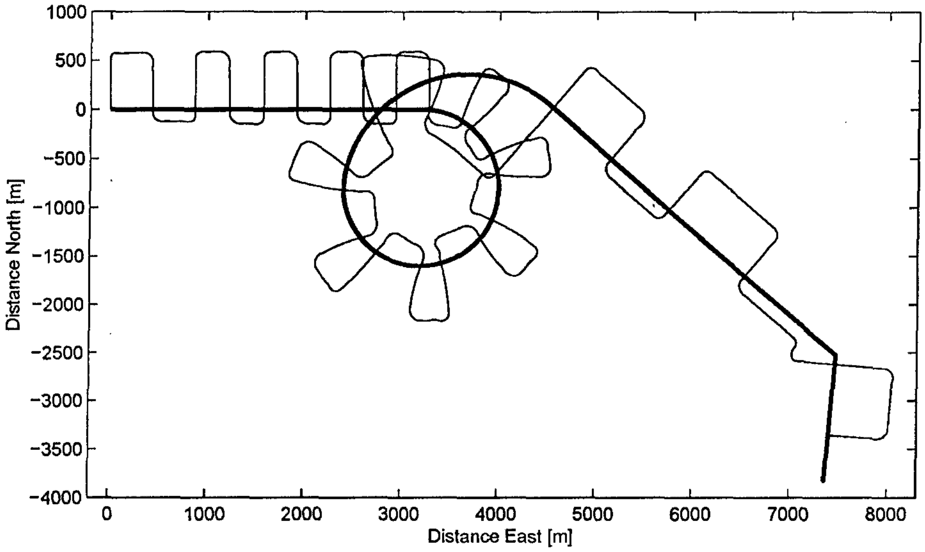


Figure 3.5: UAV's flight pattern following a target changing course using the square wave pattern

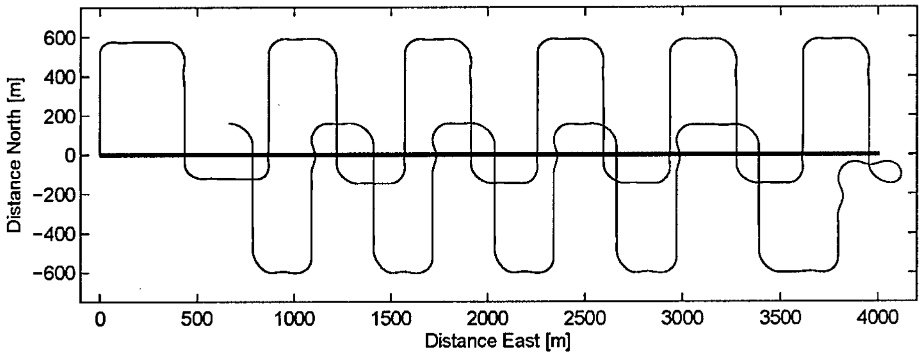


Figure 3.6: UAV's flight pattern for a course reversal using the square wave pattern

3.4 *Resetting*

Resetting is accomplished by defining s (which determines location on the pattern) equal to zero after each complete pattern according to Equation 3.14 (see Figure 3.2).

$$s = \begin{cases} s & s \leq \sum_{i=1}^{n} s_i \\ 0 & s > \sum_{i=1}^{n} s_i \end{cases} \tag{3.14}$$

3.5 *Advantages and Disadvantages*

The main advantage of this pattern is that it is capable of tracking the fastest moving target of all the methods. As stated earlier, with no wind, this pattern can track a target with a speed ratio of about 1.3:1. This is good in and of itself, but it could be improved by increasing the arbitrarily set $500m$ maximum lengths of segments a and b, although it would be a tradeoff with distance from the target.

This pattern also has the benefit of being the most simple method for adjusting the position of the UAV with respect to the target. It is just a matter of adjusting the magnitude of the a and $b1$ segments.

The big drawback to this method it that it is the only pattern that has a lower limit to the target speed it is capable of tracking. This makes it impossible for this method to be used by itself in case the target stops moving. Another disadvantage is the complexity of the feedback due to the coordinate transformation required for a turning target.

Chapter 4

CIRCULAR PATTERN

The circular pattern is different in that it is the one method that does not have a defined changing path. The UAV is simply directed to fly a path of constant curvature and the adjustments made for following the target are done purely by changes to the Serret-Frenet variables y_s and χ. There are two forms of this pattern. The first we refer to as the 'independent' circular pattern and the second as the 'integrated' circular pattern.

Note: Unlike the other two patterns where target location is continually updated, for the circular pattern we designed the algorithm such that it updates the target position whenever the target has moved 50m (a more realistic scenario). For this reason the plots using the circular pattern do not look smooth.

4.1 Independent Circular Pattern

The simplest method of all is the independent circular method. This pattern essentially does not use the Serret-Frenet method since $y_{s_{new}}$ is not fed back like normal (see Simulink model in Appendix B.3). Instead, y_s is directly defined by Equation 4.1. Unlike all of the other methods this is constantly updated as opposed to being updated at certain points during the pattern.

$$y_s = r - \sqrt{(x_t - x)^2 + (y - y_t)^2} \tag{4.1}$$

where y_s is limited by:

$$-(r-1) \leq y_s \leq (r-1). \tag{4.2}$$

The course is still calculated using Equation 4.3 because this method ignores any Δs (essentially leaving out in which direction the Δy_s lies).

$$\chi = s\kappa \tag{4.3}$$

Because of this, y_s is the only variable that changes when the target moves, so the algorithm just tells the UAV how far it is from the target but not in which direction. This greatly limits the capability of this method. Empirically it can be shown that it only works accurately for target speeds less than $7m/s$ or a ratio of 3.6:1 (these values could be improved with increased updates of target location or with an increased radius of the pattern, but not significantly). Below this limit however, this method is a very simple solution to the problem and works well. Figure 4.1 shows an example of this technique. The UAV can actually track a target moving a little faster than the stated limit but the target gets outside the circle. Another issue is that it will be strongly affected by wind. Because of this, like the square wave method, this pattern could never be used by itself. These two methods could however be used in combination. Section 4.4 discusses the standard circular method being combined with the square wave pattern but the same could be done with this method with just a few minor adjustments.

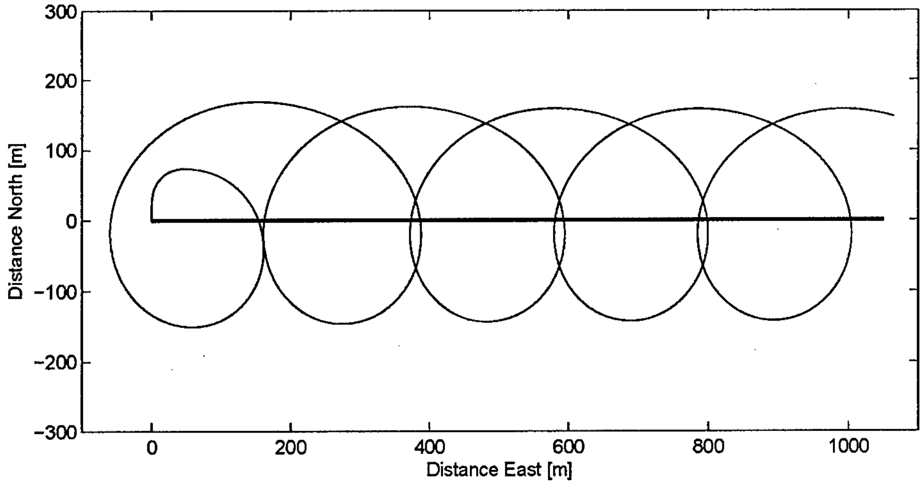


Figure 4.1: The flight pattern of the UAV using the independent circular method to track the target

4.2 Basic Pattern Movement of the Integrated Circular Pattern

The integrated circular method can follow a faster moving target with the same radius. However, improving upon the target speeds possible by the independent circular method requires a much more complex method. Figure 4.2 shows the geometry used to calculate the changes needed to move the circle with the target. These values allow for the calculations of Δs, Δy_s, and $\Delta \chi$ needed to update the Serret-Frenet path following algorithm (Equation 2.9). The distance from the UAV to the new target position is given by

$$\Gamma = \sqrt{(r + y_{s_o})^2 + S_T^2 - 2\,(r + y_{s_o})\,S_T \cos\gamma} \tag{4.4}$$

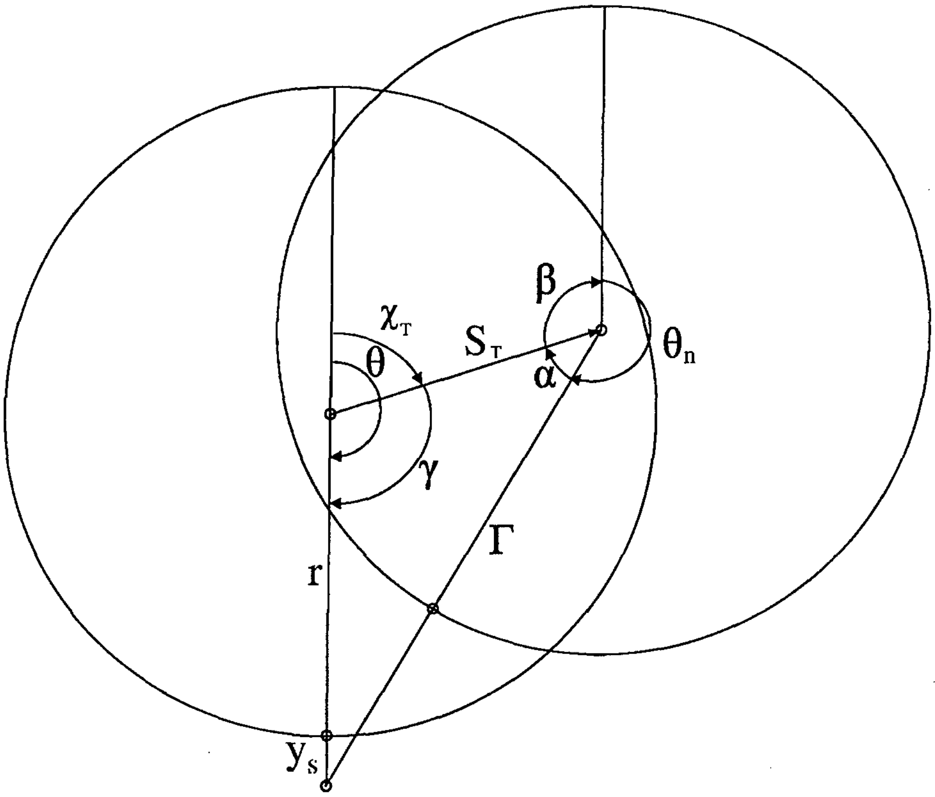


Figure 4.2: Geometry used to calculate the distances and angles necessary to move the integrated circular pattern with the movement of the target

where the angle γ is defined by:

$$\gamma = \begin{cases} |\theta - \chi_t| & |\theta - \chi_t| \leq \pi \\ 2\pi - |\theta - \chi_t| & |\theta - \chi_t| > \pi \end{cases} . \tag{4.5}$$

The angles α and β are given by Equations 4.6 and 4.7.

$$\alpha = \begin{cases} \pi - \arcsin\left(\frac{r - y_{s_o}}{\Gamma} \sin\gamma\right) & \left[(r + y_{s_o})^2 - S_T^2 - \Gamma^2\right] > \pi \\ \arcsin\left(\frac{r - y_{s_o}}{\Gamma} \sin\gamma\right) & \left[(r + y_{s_o})^2 - S_T^2 - \Gamma^2\right] \leq \pi \end{cases} \tag{4.6}$$

$$\beta = |\pi - \chi_t| \tag{4.7}$$

These are used in calculating the change in arclength position along the desired path according to Equation 4.8.

$$\Delta s = (\theta_n - \theta)\, r \tag{4.8}$$

where θ_n is determined by the following logic.

```
if $\chi_t > \pi$
    if $\theta - \chi_t > \pi$
        $\theta_n = \beta + \alpha$
    else
        if $\theta - \chi_t > 0$
            if $(S_t - r - y_{s_o}) \sin\theta > 0$
                $\theta_n = 2\pi - |\beta - \alpha|$
            else
                $\theta_n = |\beta - \alpha|$
        else
            if $\chi_t - \theta > \pi$
                if $(S_t - r - y_{s_o}) \sin\theta > 0$
                    $\theta_n = 2\pi - |\beta - \alpha|$
                else
                    $\theta_n = |\beta - \alpha|$
            else
                $\theta_n = \beta + \alpha$
```

```
else
  if $\theta - \chi_t > \pi$
    if $(S_t - r - y_{s_o})\sin\theta > 0$
      $\theta_n = 2\pi - |\beta - \alpha|$
    else
      $\theta_n = |\beta - \alpha|$
  else
    if $\theta - \chi_t > 0$
      $\theta_n = 2\pi - \beta - \alpha$
    else
      if $(S_t - r - y_{s_o})\sin\theta > 0$
        $\theta_n = 2\pi - |\beta - \alpha|$
      else
        $\theta_n = |\beta - \alpha|$
```

The new heading is then given by

$$\chi = \kappa(s + \Delta s). \tag{4.9}$$

The change in normal distance from the desired path is given by Equation 4.10. This is currently an open issue since Δy_s is a constantly increasing number due to the fact that Δy_{s_o} is added in each time Δy_s is calculated. Therefore, more work will need to be done in order to keep this as a finite value over long periods of time.

$$\Delta y_s = \Delta y_{s_o} + \mu \tag{4.10}$$

where

```
if $|y_s + r - \Gamma| > r - 1$
  if $y_s + r - \Gamma > 0$
    $\mu = (r - 1) - y_s$
  else
```

```
    $\mu = -(r-1) - y_s$
else
    $\mu = r - \Gamma$
```

and the total y_s added into the Serret-Frenet operator is limited by:

$$-(r-1) \leq y_s \leq (r-1). \tag{4.11}$$

4.3 Feedback

Feedback for the integrated circle is done at two points around the circle ($\chi_s = 0°$ and $\chi_s = 90°$) simply by adjusting y_s using Equations 4.12 through 4.14.

$$y_{s_f} = y_{s_{x_f}} + y_{s_{y_f}} \tag{4.12}$$

$$y_{s_{x_f}} = y_{s_{x_{fo}}} + \nu \tag{4.13}$$

where ν is defined by:

```
if $|y_s - x_t + x + r| > r - 1$
  if $y_s - x_t + x + r > 0$
    $\nu = (r-1) - y_s$
  else
    $\nu = -(r-1) - y_s$
```

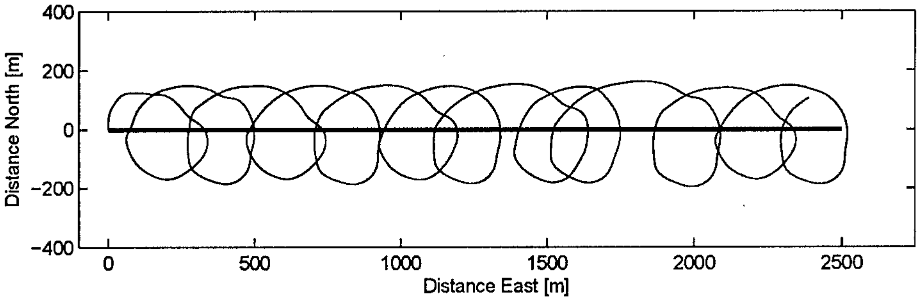


Figure 4.3: UAV's flight pattern following a target using the integrated circular pattern

```
else
  $\nu = x - x_t + r$
```

and

$$y_{s_{y_f}} = y_{s_{y_{fo}}} + \xi \tag{4.14}$$

where ξ is defined by:

```
if $|y_s - y + y_t + r| > r - 1$
  if $y_s - y + y_t + r > 0$
    $\xi = (r-1) - y_s$
  else
    $\xi = -(r-1) - y_s$
else
  $\xi = -y + y_t + r$
```

and the initial value of $(y - y_t) = r$.

This feedback ensures that the target's path is centered in the circular pattern as seen in Figure 4.3.

4.4 Simulation

Because the circle has infinite rotational symmetry, unlike the square wave pattern, no coordinate transformation is required when the target changes direction. Course correction is taken care of in the same manner that the pattern is moved when it is move in a straight line. An example of the target changing direction is shown if Figure 4.4.

There is no minimum speed for this method and again the maximum speed cannot be determined mathematically. Empirically we determined the maximum target speed to be about $12m/s$ or a ratio of 2.1:1. This is based on the target position being updated each time it has moved 50m and the radius of the integrated circular pattern set at 150m. The maximum target speed increases when the target position is updated more frequently or when the radius of the pattern is increased. Figure 4.5 shows the case when target position is updated each time it has moved 50m and the radius of the circular pattern set at 150m

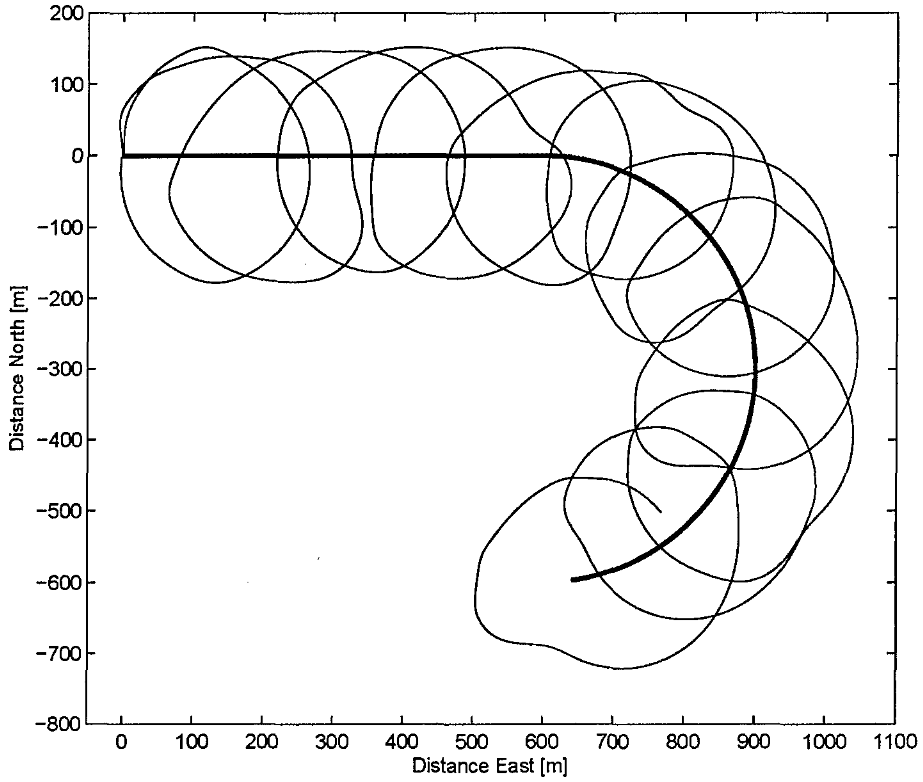


Figure 4.4: UAV's flight pattern following a turning target using the integrated circular pattern

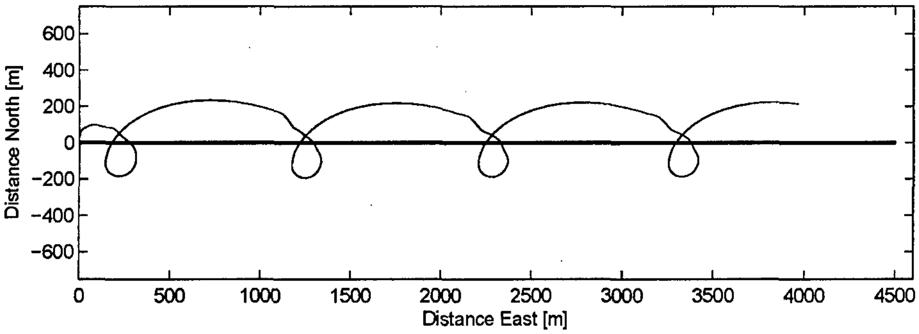


Figure 4.5: UAV's flight pattern following a target traveling at 15m/s using the integrated circular pattern with a radius of 150m being updated each time the target has moved 50m

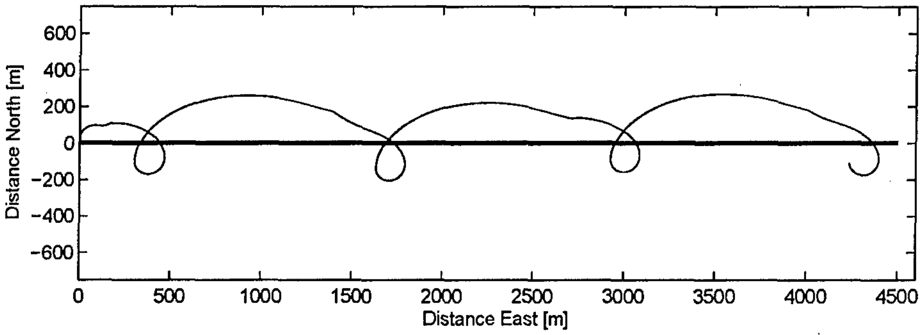


Figure 4.6: UAV's flight pattern following a target traveling at 15m/s using the integrated circular pattern with a radius of 150m being updated each time the target has moved 1m

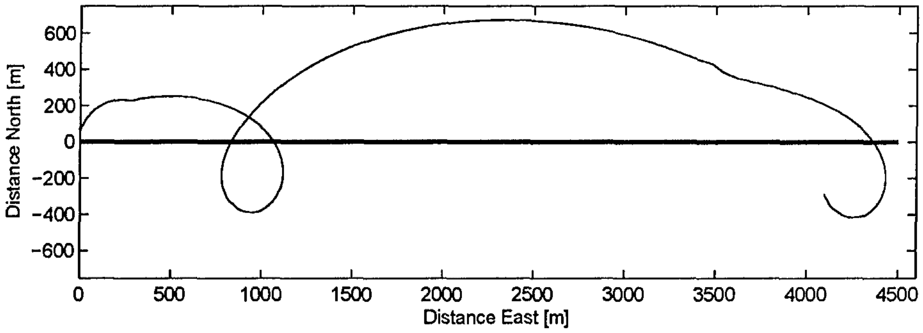


Figure 4.7: UAV's flight pattern following a target traveling at 15m/s using the integrated circular pattern with a radius of 500m being updated each time the target has moved 50m

for a target moving at $15m/s$. Figure 4.6 shows what happens when the target position is updated every meter. Figure 4.7 shows the case when the target position is updated every 50m and the radius of the pattern is increased to 500m. As can be seen, in these last two cases, the capability to follow the target can be improved but the cost is either better technology is required or increased distance from the target.

Because the square wave pattern has a minimum target velocity that it can follow and the circular pattern does not, yet cannot reach the same maximum speed that the square wave pattern can, it is a logical conclusion to combine the two methods. This can be done through a few simple modifications to the code. A switching point based on the ratio of airspeed to target velocity is used with a hysteresis to avoid frequent switching. Also the

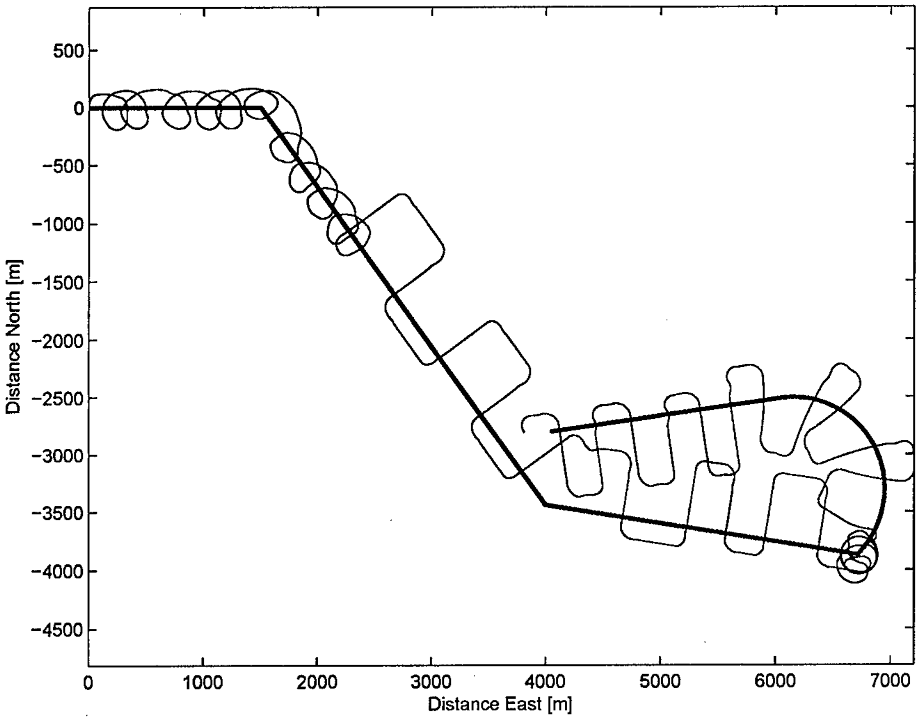


Figure 4.8: Switching between the integrated circular pattern and the square wave pattern

values of s and y_s must be reset each time a switch is made. Figure 4.8 shows an example of the UAV switching back and forth between the square wave pattern and the integrated circular pattern.

4.5 Advantages and Disadvantages

As the name implies, the independent circular method has the advantage of being extremely simple. This simplicity however, comes at the cost of having a relatively low maximum speed and more importantly getting completely off track when the target travels too fast for it to keep up.

One of the advantages of the integrated circular method is that it allows for longer time between updates of the target position than the other methods allow. This is because it only needs to keep the target within its own radius for it to stay within view.

Another advantage is that it has simpler feedback than square wave because no coordinate transformation is required for changing course, although it does work with coordinate transformation (as it does when it is combined with the square wave method).

Another aspect of this method that simplifies it compared with the others is that it has a constant κ (making the whole pattern equivalent to one segment of one of the other patterns).

The main disadvantage of this method is the complexity of calculating all of the angles required due to the number and size of all of the if-then-else statements required.

The other drawbacks to this method deal with the calculation of Δy_s. First, there is the limit that the magnitude of y_s be less than the radius of the circle. This limits the target speed that this method is capable of following. And second is the issue of the value of Δy_s going off to infinity as time goes to infinity, although this is a problem that should be able to be resolved.

Chapter 5

STANDOFF PATTERN

The main purpose of the standoff pattern is to remain on one side of the target, i.e. to stay between the target and the sun. It is a more specialized pattern and therefore both more complex and limited than the other two patterns.

5.1 Basic Pattern

The basic pattern is made up of 8 segments as seen in Figure 5.1 and described in Table 5.1. It is designed such that segments 4 and 8 are parallel to and equidistant from the sides of

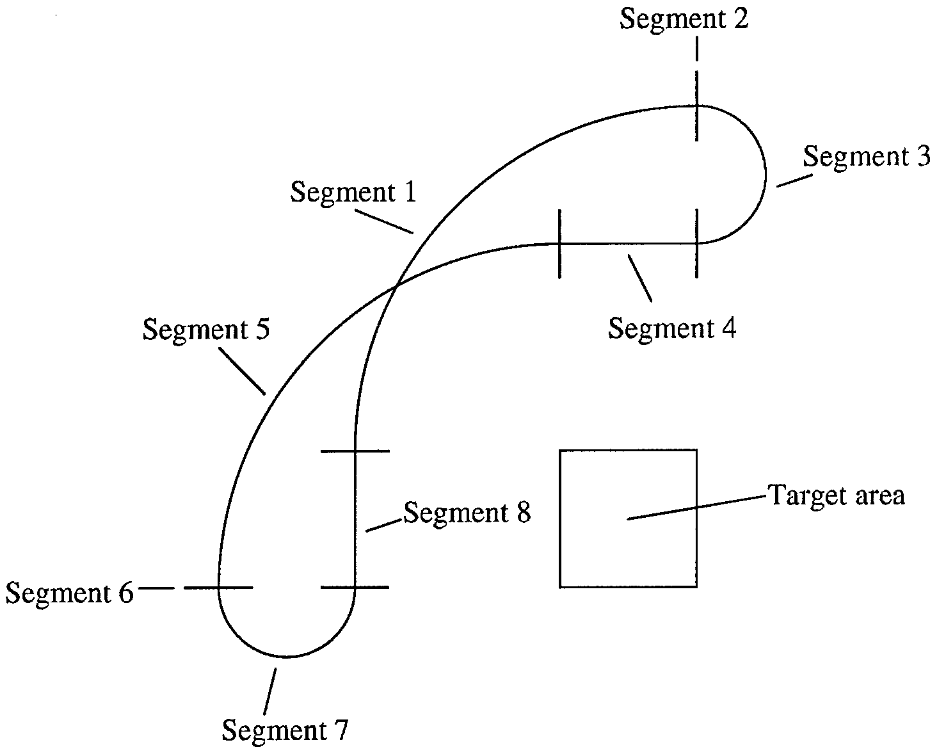


Figure 5.1: Basic unit of the standoff pattern

Table 5.1: Segment definitions for the standoff pattern

Segment	Description	Nominal s Value
s_1	Quarter circle of radius 250m centered around the upper right corner of the target area	392.7
s_2	Straight segment of variable length	1
s_3	Half circle of variable radius (nominally 50m)	157.1
s_4	Straight segment of variable length	101
s_5	Quarter circle of radius 250m centered around the lower left corner of the target area	392.7
s_6	Straight segment of variable length	1
s_7	Half circle of variable radius (nominally 50m)	157.1
s_8	Straight segment of variable length	101

the square target area (ideally with the target at the center). This is similar to the standoff pattern already developed for stationary targets [17] but with the addition of segments 2 and 6 to allow the pattern to be more easily modified for movement. Each segment has an initially defined curvature (κ) and length s_n.

κ (and therefore s_n) for segments 3 and 7 vary depending on the wind in order to ensure that the curvature can be flown by the UAV. We assume the wind is such that UAV can always make a turn of curvature $\kappa = 1/250$ for segments 1 and 5. The lengths of segments 2, 4, 6, and 8 (the straight segments) vary to allow for the changes in segments 3 and 7 and to move the pattern with the movements of the target.

κ is determined in the same manner that it was for the square wave path (Equation 3.4) except here we used $\phi = 40°$. V_c is also determined similar to Equation 3.5.

$$V_{c_1} = V_a + |W| \cos\left(\arctan\frac{W_y}{W_x} - \pi\right) \tag{5.1}$$

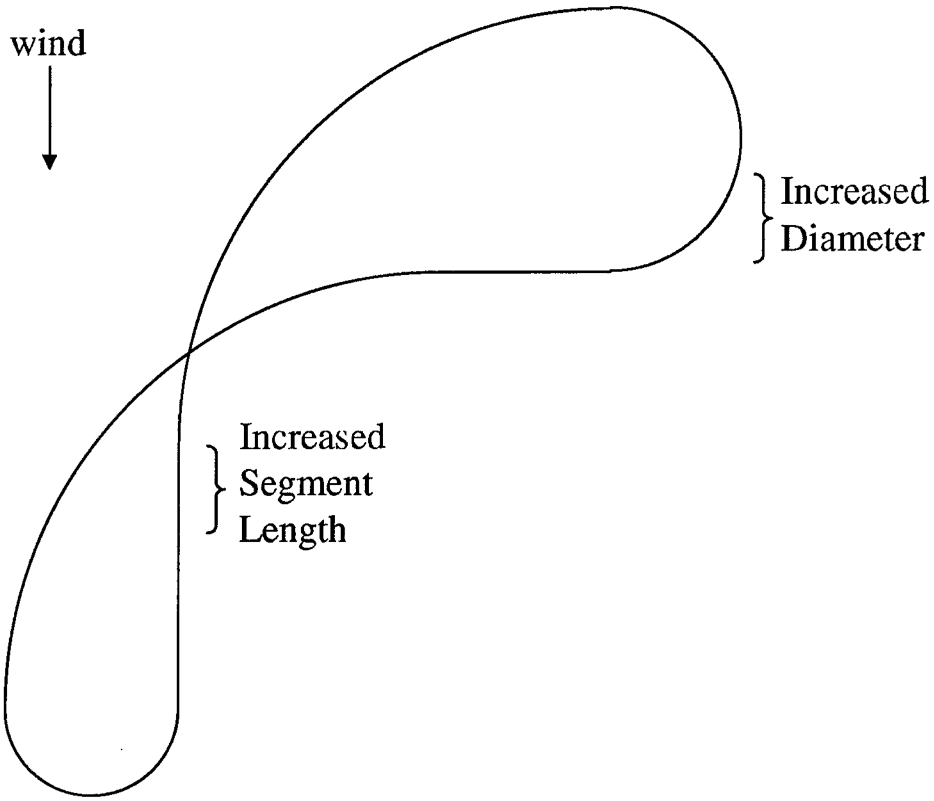


Figure 5.2: Basic unit of the standoff pattern in winds from the North

$$V_{c_2} = V_a + |W| \cos\left(\arctan \frac{W_y}{W_x} - \frac{\pi}{2}\right) \tag{5.2}$$

Another difference is that here a limit is put on κ such that $\kappa \leq 0.02$ so the radii of the 180° turns in segments 3 and 7 are never less than 50m. Figure 5.2 shows an example of the changes made on the basic pattern to account for wind speed.

The lengths of the straight segments are determined by several factors. Segments 2, 4, 6, and 8 (which we will also refer to as segments a, b, c, and d respectively) each have an initial length that corresponds with the situation where the target is not moving, there is no wind, and there are no perturbations to the system. These lengths respectively are: 1m, 101m, 1m, and 101m. The reason for the odd 1m lengths is that the code is unable to handle a segment of length zero. The next component in the segment length is the target's velocity. To account for this, the segments are adjusted by the following equations (the

UAV's velocity is accounted for in T_p).

$$a = \begin{cases} V_t T_p \cos(\chi_t) - 100 & V_t T_p \cos(\chi_t) \geq 100 \\ 0 & V_t T_p \cos(\chi_t) < 100 \end{cases} \tag{5.3}$$

$$b = \begin{cases} -100 & V_t T_p \cos(\chi_t) \geq 100 \\ -V_t T_p \cos(\chi_t) & V_t T_p \cos(\chi_t) < 100 \end{cases} \tag{5.4}$$

$$c = \begin{cases} V_t T_p \sin(\chi_t) - 100 & V_t T_p \sin(\chi_t) \geq 100 \\ 0 & V_t T_p \sin(\chi_t) < 100 \end{cases} \tag{5.5}$$

$$d = \begin{cases} -100 & V_t T_p \sin(\chi_t) \geq 100 \\ -V_t T_p \sin(\chi_t) & V_t T_p \sin(\chi_t) < 100 \end{cases} \tag{5.6}$$

The third component in determining the segment lengths is to account for the variable radii of segments 3 and 7. Segments 4 and 8 are adjusted according to Equations 5.7 and 5.8.

$$b_w = 2\,(r_2 - 50) \tag{5.7}$$

$$d_w = 2\,(r_1 - 50) \tag{5.8}$$

The final component of segment lengths is feedback.

5.2 Feedback

To start with we tried using y_s feedback (like was used in the circular pattern) at the start of segments 1 and 5. We determined that y_s feedback worked better (or only worked) with longer segments because there are problems if the UAV reaches the end of a segment and the value of y_s is still relatively large. Segments 1 and 5 are the only reliably long segments in this pattern so we could only use this feedback at those two locations. The limitation of this technique is that the greatest y_s feedback allowed must be less than the radius of the segment (in this case it has to be less than the desired distance from the target which is 200m compared with the radius of the segments which is 250m). Because of this, it is greatly limited by speed. Figure 5.3 shows this method simulated at a slow enough speed that the UAV can keep up with the target. As can be seen it does a good job of staying

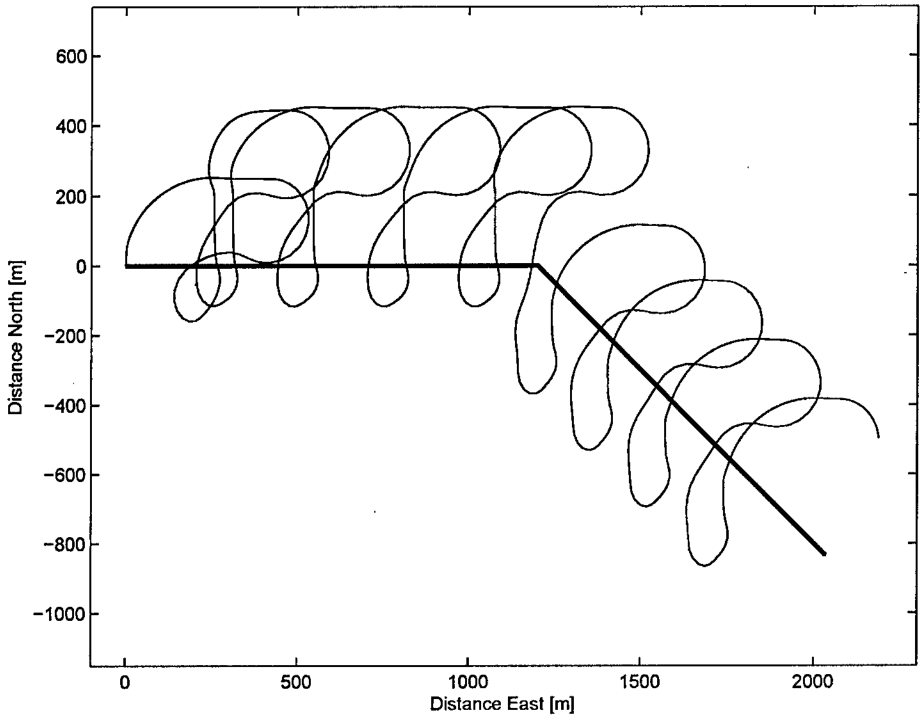


Figure 5.3: UAV's flight path when using the standoff pattern and only using y_s feedback

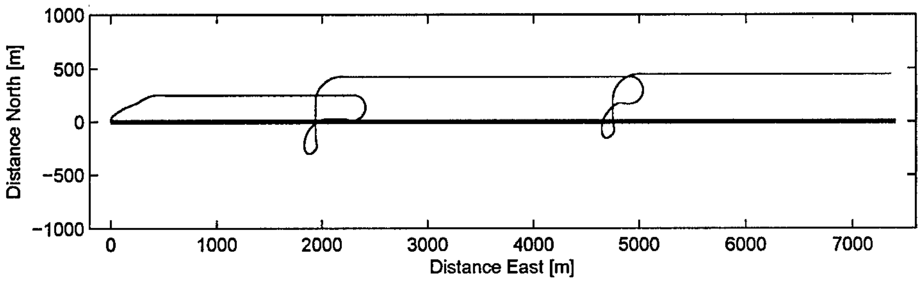


Figure 5.4: UAV's flight path when using the standoff pattern and only using abcd feedback to follow a target moving at a relatively high velocity in a straight line

with the target. However, because of the significance of the speed limitation, we decided that y_s feedback is not the best option for this pattern and used a similar method to the one we used with the square wave pattern by adjusting segments 2, 4, 6, and 8. This method works well and can keep up with a target moving at relatively high speeds (see Figure 5.4) as long as the target is moving parallel to either of the two sets of straight segments in the pattern (in the direction of one of the four compass points as the pattern is drawn in Figure 5.1) since those are the directions that can be changed by adjusting the segment lengths. It does not work well at all at high speeds for target motion at some angle. As can be seen in Figure 5.5, the UAV still tracks the target but it is often far from it and in many cases no longer stays to one side of the target. It turns out that the best method is to use a combination of the y_s feedback and this second method (that we call abcd feedback). Even so, it is still very limited by speed. Figure 5.6 shows the path of the UAV using this combination method of feedback. Abcd feedback is calculated using the following equations (the $\frac{1}{2}$ in front of each equation is due to the fact that the job of feedback is split between

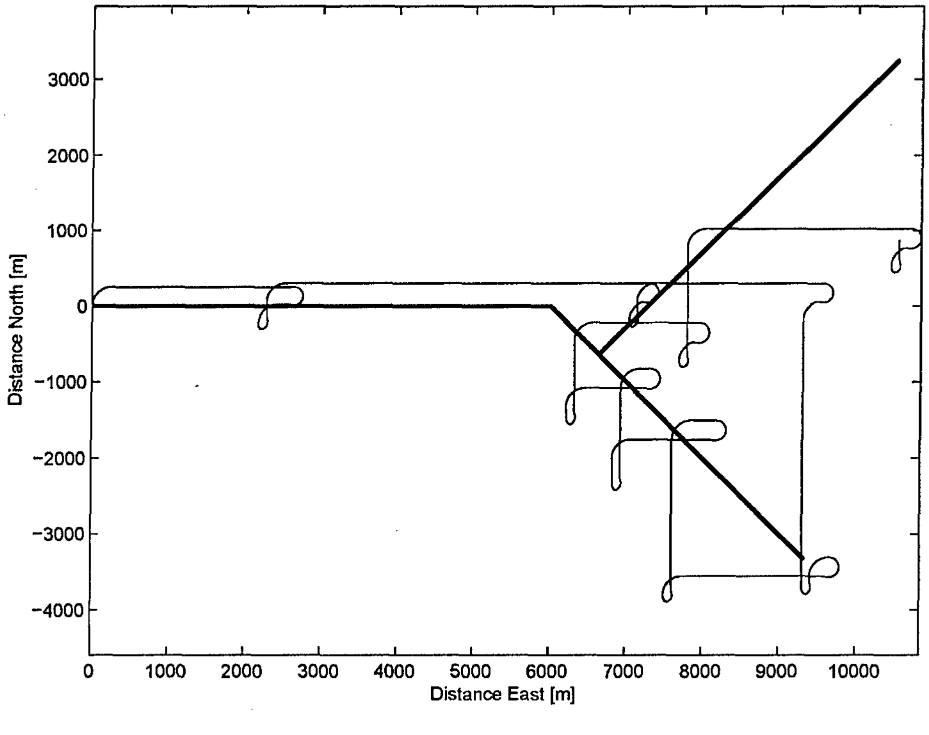


Figure 5.5: UAV's flight path when trying to follow a target moving at relatively high speed and at an angle with the standoff pattern

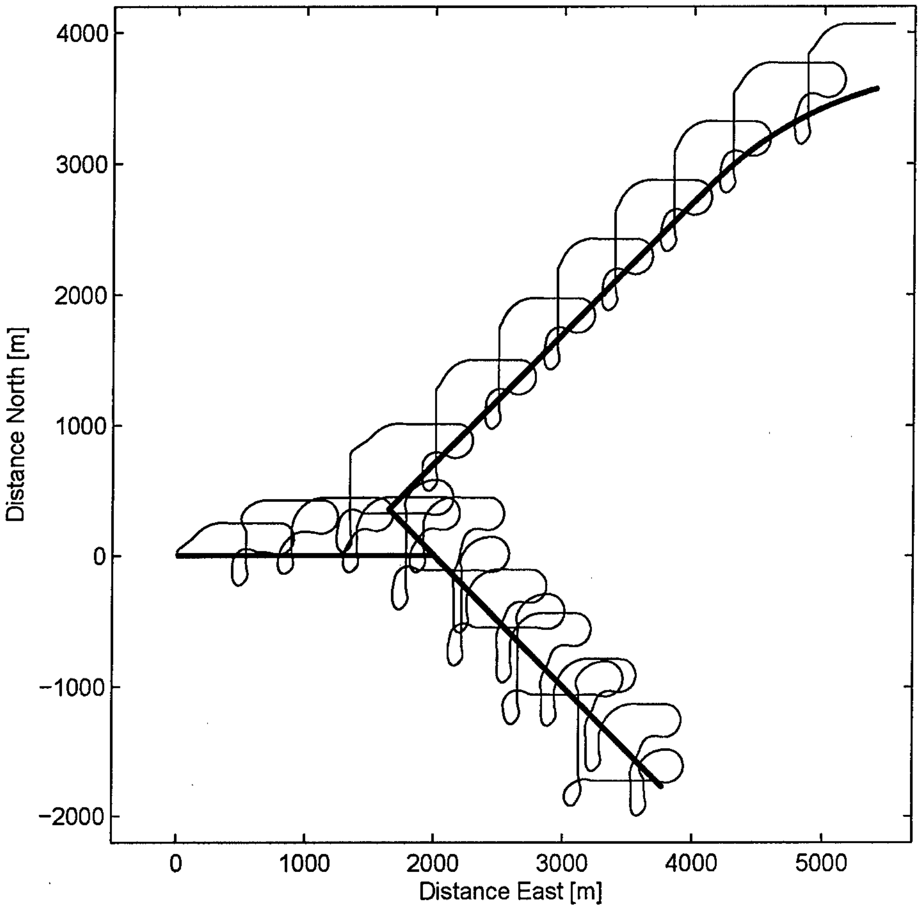


Figure 5.6: UAV's flight path when using both y_s and abcd feedback

the y_s method and the abcd method).

$$a_f = \frac{1}{2}\begin{cases} x_t - x - 200 & x_t - x - 200 > 0 \\ 0 & x_t - x - 200 \leq 0 \end{cases} \tag{5.9}$$

$$b_f = -\frac{1}{2}\begin{cases} 0 & x_t - x - 200 > 0 \\ x_t - x - 200 & x_t - x - 200 \leq 0 \end{cases} \tag{5.10}$$

$$c_f = \frac{1}{2}\begin{cases} y - y_t - 50 & y - y_t - 50 > 0 \\ 0 & y - y_t - 50 \leq 0 \end{cases} \tag{5.11}$$

$$d_f = -\frac{1}{2}\begin{cases} 0 & y - y_t - 50 > 0 \\ y - y_t - 50 & y - y_t - 50 \leq 0 \end{cases} \tag{5.12}$$

The total segment lengths can now be determined. They are given by Equations 5.13 through 5.20

$$\begin{aligned} s_1 &= constant \\ &= \frac{\pi r}{2} \\ &= \frac{\pi 250}{2} \\ &= 392.699 \end{aligned} \tag{5.13}$$

$$s_2 = 1 + az^{-1} + a_f \tag{5.14}$$

$$s_3 = \frac{\pi}{\kappa_1} \tag{5.15}$$

$$s_4 = 101 + bz^{-1} + b_f + b_w \tag{5.16}$$

$$s_5 = s_1 \tag{5.17}$$

$$s_6 = 1 + cz^{-1} + c_f \tag{5.18}$$

$$s_7 = \frac{\pi}{\kappa_2} \tag{5.19}$$

$$s_8 = 101 + dz^{-1} + d_f + d_w \tag{5.20}$$

where z^{-1} is a one time step delay. This is needed in order to avoid circular logic in the calculations since the time it takes for the UAV to complete one complete pattern (T_p) is used to calculate a, b, c, and d and is given by:

$$T_p = \sum_{i=1}^{n} \frac{s_i}{V_{c_i}} \tag{5.21}$$

where n is the number of segments in the pattern and

$$V_{c_i} = V_a + |W| \cos\left(\arctan \frac{W_y}{W_x} - \chi_i\right) \tag{5.22}$$

where χ_i is the target's heading for the ith segment (the average χ for the circular segments) and s_i is given by Equations 5.13 through 5.20.

y_s feedback is not involved in determining the segment lengths. It simply adjusts one of the Serret-Frenet parameters to, in essence, tell the UAV that it is off course and thus move the pattern. The value added to the Serret-Frenet parameter is given by:

$$\Delta y_s = \Delta y_{s_o} + y_{s_f} \tag{5.23}$$

where

$$y_{s_f} = y_{s_{x_f}} + y_{s_{y_f}}. \tag{5.24}$$

The two components of this equation are calculated similarly (again the $\frac{1}{2}$s in front of some of the equations are due to the fact that the job of feedback is split between the y_s method and the Abcd method).

$$y_{s_{x_f}} = y_{s_{x_{f_o}}} + \omega \tag{5.25}$$

where

```
if $\left|y_s + \frac{1}{2}(y - y_t - 200)\right| > D_d - 1$
  if $y_s + \frac{1}{2}(y - y_t - 200) > 0$
    $\omega = (D_d - 1) - y_s$
  else
    $\omega = -(D_d - 1) - y_s$
else
  $\omega = \frac{1}{2}(y - y_t - 200)$
```

and

$$y_{s_{y_f}} = y_{s_{y_{f_o}}} + \lambda \tag{5.26}$$

where

```
if $\left|y_s + \frac{1}{2}(x_t - x - 200)\right| > D_d - 1$
  if $y_s + \frac{1}{2}(x_t - x - 200) > 0$
    $\lambda = (D_d - 1) - y_s$
  else
    $\lambda = -(D_d - 1) - y_s$
else
  $\lambda = \frac{1}{2}(x_t - x - 200)$
```

5.3 *Advantages and Disadvantages*

The main advantage of this pattern is its ability to remain to one side of the target. This allows the camera to get the best pictures of the target based on the location of the sun. It would also keep an enemy target from getting a good view of the UAV if it had to look towards the sun in order to see it.

Another one of the advantages of this pattern, similar to the circular pattern, is that no coordinate transformation is needed to turn the pattern. This time, however, it is because the standoff pattern is designed to remain on one side of the target and therefore one would not want to rotate the pattern. Because of this, once again, no changes are needed when the target makes a turn, the segment length calculations and the feedback methods automatically correct for this.

A third advantage is that, again like the circular pattern, there is no minimum speed for this pattern.

However, one of the disadvantages is that this pattern has the lowest maximum speed taking into account that the target can move in any direction (when the target moves parallel to either set of straight segments in the standoff pattern, the maximum speed is comparable to the maximum speed of the square wave pattern). A numerical maximum cannot be determined for several reasons. One is that unlike the square wave pattern, in this pattern there is no limit put on the length of segments 2, 4, 6, and 8. Another is that besides taking into account the wind, like was done in Table 3.2, the target's course would also have to be taken into account. Finally, like the circular pattern, y_s feedback is used. Although it can only change at two specific points in the pattern rather than at any given point like in the circular method, like with the calculations of segment lengths, its capability of keeping the pattern moving with the target depends so much on the direction of target movement. Empirically, the maximum target speed that this pattern is capable of tracking is about $5m/s$ or a speed ratio of 5:1.

A final disadvantage is that this is the most complex pattern. This is not surprising since it is the most specialized pattern. It requires 8 defined segments and utilizes two types of feedback whereas each of the other two methods only require one type of feedback.

Chapter 6

CONCLUDING REMARKS

6.1 Summary

There is a need for an autonomous UAV to be able to fly in such a manner that it can keep a target in view of a nose mounted camera. One possible answer to this problem is to have preset patterns that the UAV can fly near the target and that will change frequently with the movements of the target.

The three patterns presented in this thesis are possible solutions to this problem each with its own advantages and disadvantages. All of them have been simulated using the Aerosonde UAV model in Simulink and shown to work under certain conditions. Although the circle is the most basic geometric shape of the three patterns, implementing it is actually more complex than the square wave pattern but it is good at low speeds (down to zero) and can go faster than the standoff pattern. The independent circular pattern is very easy to implement but has the most problems with a target moving at higher speeds. The square wave pattern is the most practical method to implement and can achieve the greatest speeds but it is limited at lower speeds and cannot come to a stand still. The standoff pattern is limited the most in the speeds it is capable of tracking but it is still useful for its purpose of staying to one side of the target. Some combination of these methods (or similar ones), such as the combination of the square wave and circular patterns discussed in Chapter 4, would be the best solution to the problem.

6.2 Future Work

The next step that is needed (besides fixing the issue with Δy_s) is to convert the Simulink models (shown in Appendix B) into C-code to be used on an actual UAV. Then the camera needs to be integrated such that it 'knows' in which direction to point, based on where in the patten the UAV currently is and the last known position and velocity of the target. A method of finding the target will also need to be integrated into the algorithm. This could be accomplished with a search pattern. This search pattern could also be used in the case that the UAV loses sight of the target. Ultimately, all of this could be used by a group of UAVs working together in combination with an image processing algorithm to update target position and velocity using images from onboard cameras.

BIBLIOGRAPHY

[1] Aerosonde. web-site. http://www.aerosonde.com/drawarticle/42.

[2] Dynamic model of aerosonde UAV. web-site. Unmanned Dynamics LLC., http://www.u-dynamics.com/aerosim/.

[3] Simulink: Simulation and model-based design. web-site. The Mathworks: http://www.mathworks.com/products/simulink/?BB=1.

[4] Unmanned aerial vehicles roadmap 2002-2027. DoD memorandum, Washington, DC, December 2002.

[5] Juan Carlos Rubio amd Juris Vagners and Rolf Rysdyk. Adaptive path planning for autonomous UAV oceanic search missions. In *AIAA 2004-6228*, Chicago, IL, September 2004.

[6] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association.* Academic Press, Boston, MA, 1988.

[7] Anouck R. Girard, Adam S. Howell, and J. Karl Hedrick. Border patrol and surveillance missions using multiple unmanned aerial vehicles. Submitted to IEEE Conference on Decision and Control, 2004.

[8] Insitu. Cooperative tracking of moving targets by a team of autonomous UAVs. Proposal. F045-011-0033.

[9] M. Isard and A. Blake. Condensation-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 28(1):5–28, 1998.

[10] Jusuk Lee, Rosemary Huang, Andrew Vaughn, Xiao Xiao, and J. Karl Hedrick. Strategies of path-planning for a UAV to track a ground vehicle. In *Proceedings of the 2nd annual Autonomous Intelligent Networks and Systems Conference*, Menlo Park, CA, June 2003. AINS.

[11] Tad McGeer. Trajectory-following course tracker. Insitu Proprietary Work Note, September 2002.

[12] K. Okuma, J.J. Little, and D. Lowe. Automatic acquisition of motion trajectories: Tracking hockey players. In *Proceedings of SPIE Internet Imaging V*, pages 202–213, San Jose, CA, January 2004.

[13] K.Y. Pettersen and E. Lefeber. Way-point tracking control of ships. In *Proc. 40th IEEE Conference on Decision and Control*, pages 940–945, Orlando, FL, December 2001.

[14] Anawat Pongpunwattana. *Real-Time Planning for Teams of AUtonomous Vehicles in Dynamic Uncertain Environments.* Ph.d. dissertation, University of Washington, 2004.

[15] Anawat Pongpunwattana and Rolf Rysdyk. Real-time planning for multiple autonomous vehicles in dynamic uncertain environments. *Aerospace Computing, Information, and Communicaion*, 1, December 2004.

[16] Rolf Rysdyk. UAV path following for constant line-of-sight. In *AIAA 2003-6626*, San-Diego, CA, 2003.

[17] Sebastian Stolle and Rolf Rysdyk. Flight path following guidance for unmanned air vehicles with pan-tilt camera for target observation. In *22nd IEEE: Digital Avionics Systems Conference*, Indianapolis, IN, October 2003.

Appendix A

AIRCRAFT DETAILS

A.1 Aerodynamic and Physical Parameters of the Aerosonde

The following excerpt is from the file `aerosondeconfig.m` which is read by the aerosim blockset and contains the aerodynamic parameters for the Aerosonde [2].

```
%%% AERODYNAMICS %%%
% Aerodynamic force application point (usually the aerodynamic center)[x y z]
rAC = [0.1425 0 0]; % m

%%% Aerodynamic parameter bounds %%%
% Airspeed bounds
VaBnd = [15 50]; % m/s
% Sideslip angle bounds
BetaBnd = [-0.5 0.5]; % rad
% Angle of attack bounds
AlphaBnd = [-0.1 0.3]; % rad

%%% Aerodynamic reference parameters %%%
% Mean aerodynamic chord
MAC = 0.189941; % m
% Wind span
b = 2.8956; % m
% Wing area
S = 0.55; % m^2

% ALL aerodynamics derivatives are per radian:
%%% Lift coefficient %%%
% Zero-alpha lift
CL0 = 0.23;
% alpha derivative
```

```
CLa = 5.6106;
% Lift control (flap) derivative
CLdf = 0.74;
% Pitch control (elevator) derivative
CLde = 0.13;
% alpha-dot derivative
CLalphadot = 1.9724;
% Pitch rate derivative
CLq = 7.9543;
% Mach number derivative
CLM = 0;

%%% Drag coefficient %%%
% Lift at minimum drag
CLmind = 0.23;
% Minimum drag
CDmin = 0.0434;
% Lift control (flap) derivative
CDdf = 0.1467;
% Pitch control (elevator) derivative
CDde = 0.0135;
% Roll control (aileron) derivative
CDda = 0.0302;
% Yaw control (rudder) derivative
CDdr = 0.0303;
% Mach number derivative
CDM = 0;
% Oswald's coefficient
osw = 0.75;

%%% Side force coefficient %%%
% Sideslip derivative
CYbeta = -0.83;
% Roll control derivative
CYda = -0.075;
% Yaw control derivative
```

```
CYdr = 0.1914;
% Roll rate derivative
CYp = 0;
% Yaw rate derivative
CYr = 0;

%%% Pitch moment coefficient %%%
% Zero-alpha pitch
Cm0 = 0.135;
% alpha derivative
Cma = -2.7397;
% Lift control derivative
Cmdf = 0.0467;
% Pitch control derivative
Cmde = -0.9918;
% alpha_dot derivative
Cmalphadot = -10.3796;
% Pitch rate derivative
Cmq = -38.2067;
% Mach number derivative
CmM = 0;

%%% Roll moment coefficient %%%
% Sideslip derivative
Clbeta = -0.13;
% Roll control derivative
Clda = -0.1695;
% Yaw control derivative
Cldr = 0.0024;
% Roll rate derivative
Clp = -0.5051;
% Yaw rate derivative
Clr = 0.2519;

%%% Yaw moment coefficient %%%
% Sideslip derivative
```

```
Cnbeta = 0.0726;
% Roll control derivative
Cnda = 0.0108;
% Yaw control derivative
Cndr = -0.0693;
% Roll rate derivative
Cnp = -0.069;
% Yaw rate derivative
Cnr = -0.0946;


%%% PROPELLER %%%
%Propulsion force application point (usually propeller hub) [x y z]
rHub = [0 0 0]; % m
% Advance ratio vector
J = [-1 0 0.1 0.2 0.3 0.35 0.4 0.45 0.5 0.6 0.7 0.8 0.9 1 1.2 2];
% Coefficient of thrust look-up table CT = CT(J)
CT = [0.0492 0.0286 0.0266 0.0232 0.0343 0.034 0.0372 0.0314 ...
    0.0254 0.0117 -0.005 -0.0156 -0.0203 -0.0295 -0.04 -0.1115];
% Coefficient of power look-up table CP = CP(J)
CP = [0.0199 0.0207 0.0191 0.0169 0.0217 0.0223 0.0254 0.0235 ...
    0.0212 0.0146 0.0038 -0.005 -0.0097 -0.018 -0.0273 -0.0737];
% Propeller radius
Rprop = 0.254; % m
% Propeller moment of inertia
Jprop = 0.002; % kg*m^2


%%% ENGINE %%%
% Engine rpm vector
RPM = [1500 2100 2800 3500 4500 5100 5500 6000 7000]; % rot per min
% Manifold pressure vector
MAP = [60 70 80 90 92 94 96 98 100]; % kPa

% Sea-level fuel flow look-up table fflow = fflow(RPM, MAP)
% RPM -> rows, MAP -> columns
```

```
FuelFlow = [
    31 32 46 53 55 57 65 73 82
    40 44 54 69 74 80 92 103 111
    50 63 69 92 95 98 126 145 153
    66 75 87 110 117 127 150 175 190
    83 98 115 143 148 162 191 232 246
    93 102 130 159 167 182 208 260 310
    100 118 137 169 178 190 232 287 313
    104 126 151 184 191 206 253 326 337
    123 144 174 210 217 244 321 400 408
]; % g/hr
% Sea-level power look-up table P = P(RPM, MAP)
% RPM -> rows, MAP -> columns
Power = [
    18.85 47.12 65.97 67.54 69.12 67.54 67.54 69.12 86.39
    59.38 98.96 127.55 149.54 151.74 160.54 178.13 200.12 224.31
    93.83 149.54 187.66 237.5 249.23 255.1 307.88 366.52 398.77
    109.96 161.27 245.57 307.88 326.2 351.86 421.5 491.14 531.45
    164.93 245.04 339.29 438.25 447.68 494.8 565.49 673.87 772.83
    181.58 245.67 389.87 496.69 528.73 571.46 662.25 822.47 993.37
    184.31 293.74 403.17 535.64 570.2 622.04 748.75 956.09 1059.76
    163.36 276.46 420.97 565.49 609.47 691.15 860.8 1130.97 1193.81
    124.62 249.23 417.83 586.43 645.07 762.36 996.93 1246.17 1429.42
]; % W
% Sea-level pressure and temperature at which the data above is given
pSL = 102300; % Pa
TSL = 291.15; % deg K
% Engine shaft moment of inertia
Jeng = 0.0001; % kg*m^2


%%% INERTIA %%%
% Empty aircraft mass (zero-fuel)
mempty = 8.5; % kg
% Gross aircraft mass (full fuel tank)
mgross = 13.5; % kg
```

```
% Empty CG location [x y z]
CGempty = [0.156 0 0.079]; % m
% Gross CG location [x y z]
CGgross = [0.159 0 0.090]; % m
% Empty moments of inertia [Jx Jy Jz Jxz]
Jempty = [0.7795 1.122 1.752 0.1211]; % kg*m^2
% Gross moments of inertia [Jx Jy Jz Jxz]
Jgross = [0.8244 1.135 1.759 0.1204]; % kg*m^2


%%% OTHER SIMULATION PARAMETERS %%%
% WMM-2000 date [day month year]
dmy = [13 05 2002];
```

A.2 Linearized Models

Linear models of the Aerosonde were obtained using the linearization script `aerosonde_trim.m` included with the Aerosim blockset. The flight condition input to the script was Airspeed of 25 m/s, Altitude of 1000m, fuel mass of 2 Kg, and straight and level flight. The script's output follows:

```
Finished. The trim results are:
INPUTS:
   Elevator = -0.0759
   Aileron = -0.0083
   Rudder = -0.0010
   Throttle = 0.7075
STATES:
   u =  24.96 m/s
   v =   0.01 m/s
   w =   1.41 m/s
   p =   0.00 deg/s
   q =  -0.00 deg/s
   r =   0.00 deg/s
   phi =  -0.02 deg
   theta =   3.24 deg
   psi =  -0.11 deg
   Alt = 1000.00 m
   Fuel =   2.00 kg
   Engine =   5209 rot/min
OUTPUTS:
   Airspeed =  25.00 m/s
   Sideslip =   0.02 deg
   AOA =   3.24 deg
   Bank =  -0.02 deg
   Pitch =   3.24 deg
   Heading = 359.89 deg
   Altitude =  1000.00 m

Extracting aircraft linear model...


 Longitudinal Dynamics
-----------------------
  State vector: x = [u w q theta h Omega]
  Input vector: u = [elevator throttle]
 Output vector: y = [Va alpha q theta h]
 State matrix: A =
   -0.2205    0.5420   -1.3926   -9.8121   -0.0001    0.0107
   -0.5523   -4.1290   24.4105   -0.5560    0.0010         0
    0.4370   -4.4992   -4.7625         0    0.0000   -0.0083
         0         0    1.0000         0         0         0
    0.0566   -0.9984         0   24.9997         0         0
   32.0796    1.8177         0         0   -0.0383   -3.1132

 Control matrix: B =
    0.3384         0
   -2.3488         0
  -32.4992         0
         0         0
         0         0
         0  301.2339

 Observation matrix: C =
   0.9984    0.0566         0         0         0         0
  -0.0023    0.0399         0         0         0         0
        0         0    1.0000         0         0         0
        0         0         0    1.0000         0         0
        0         0         0         0    1.0000         0

Eigenvalue: -4.4438 +/- 10.5023 i
Damping = 0.3897, natural frequency = 11.4038 rad/s, period =   0.5983 s
Eigenvalue: -3.2169
Time constant = 0.3109 s
Eigenvalue: -0.0600 +/- 0.5322 i
Damping = 0.1121, natural frequency = 0.5356 rad/s, period =  11.8050 s
Eigenvalue: -0.0007
Time constant = 1526.9361 s

Lateral-directional Dynamics
------------------------------
 State vector: x = [v p r phi psi]
 Input vector: u = [aileron rudder]
Output vector: y = [beta p r phi psi]
State matrix: A =
  -0.6382    1.4143  -24.9597    9.8121         0
  -4.2032  -20.6894    9.9576         0         0
   0.6818   -2.6836   -1.0407         0         0
        0    1.0000    0.0567   -0.0000         0
        0         0    1.0016   -0.0000         0

Control matrix: B =
  -1.3638    3.4810
-119.7422    2.1545
  -4.7213  -21.9948
        0         0
        0         0

Observation matrix: C =
   0.0400         0         0         0         0
```

```
        0    1.0000         0         0         0
        0         0    1.0000         0         0
        0         0         0    1.0000         0
        0         0         0         0    1.0000

Eigenvalue: -19.8687
Time constant = 0.0503 s
Eigenvalue: -1.2766 +/- 5.8249 i
Damping = 0.2141, natural frequency = 5.9631 rad/s, period =   1.0787 s
Eigenvalue: 0.0535
Time constant = -18.6855 s
```

Appendix B

SIMULINK MODELS

B.1 Simulink

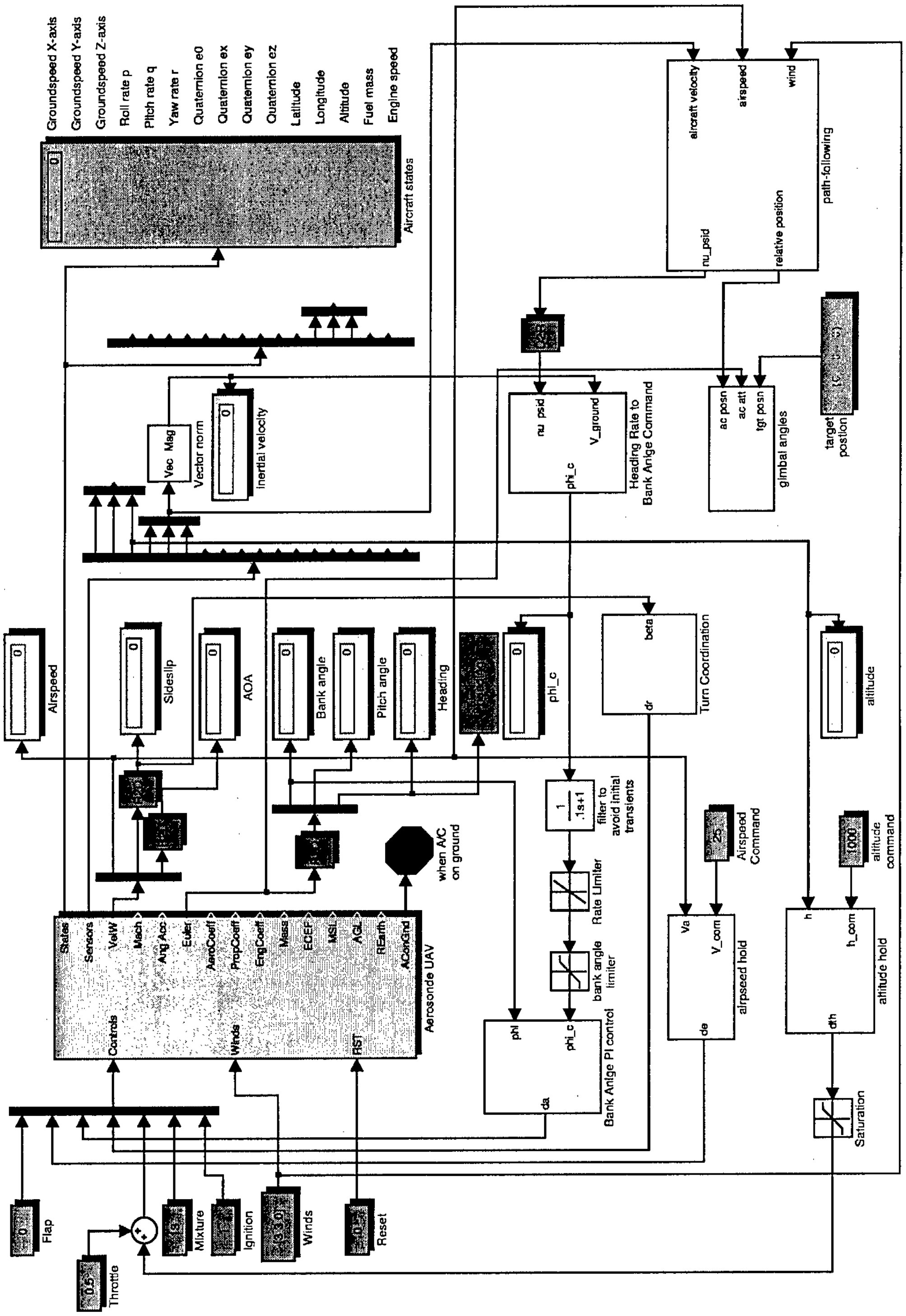
B.1.1 Description from MathWorks [3]

Simulink is a platform for multidomain simulation and Model-Based Design of dynamic systems. It provides an interactive graphical environment and a customizable set of block libraries that let you accurately design, simulate, implement, and test control, signal processing, communications, and other time-varying systems.

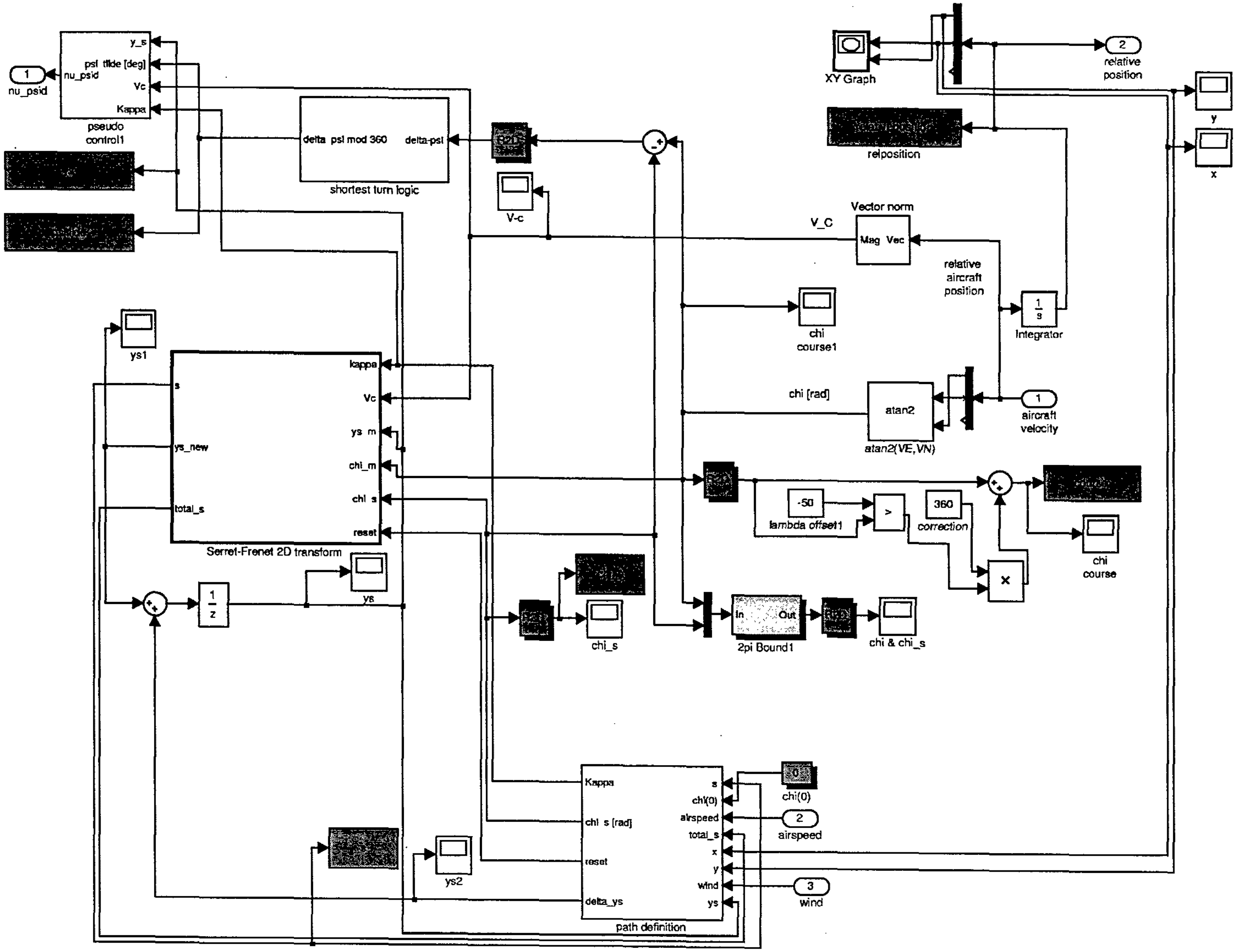
B.1.2 Settings

For my simulations in Simulink, I used the Runge-Kutta solver with a fixed step size of 0.02 seconds.

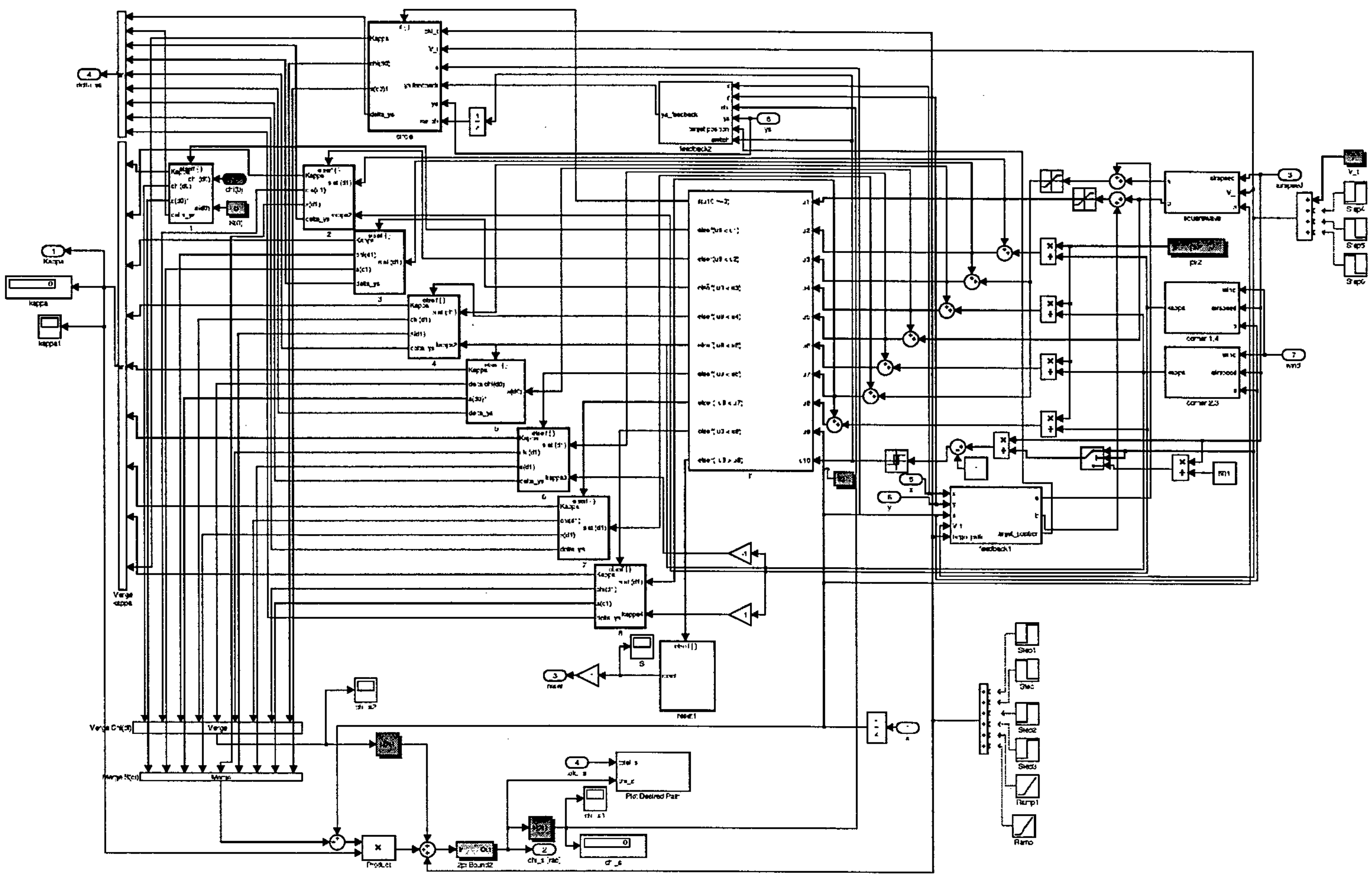
B.2 Simulink Model for the Square Wave and Circular Patterns



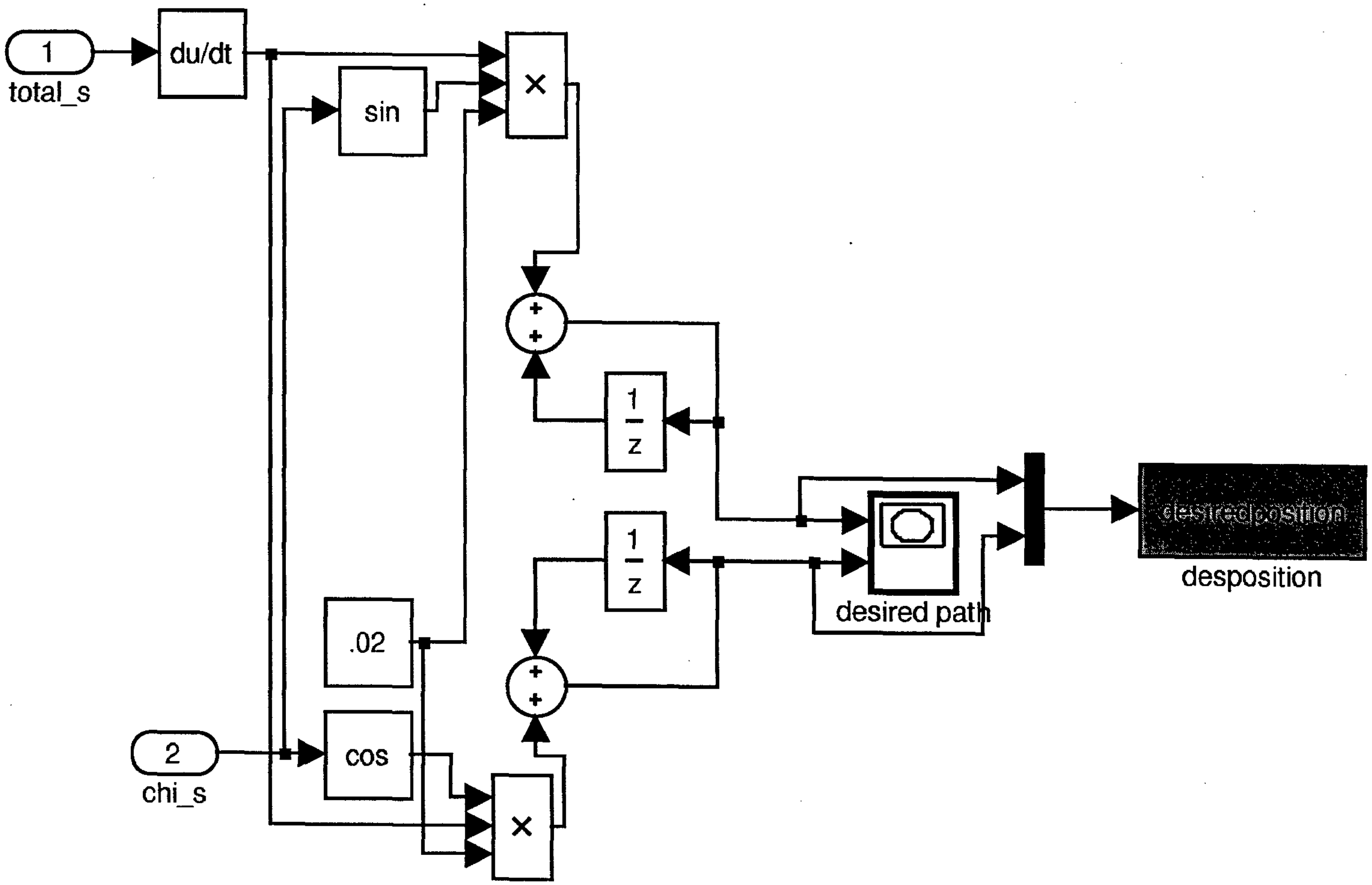
Main Aerosonde Simulink model for the square wave and circular patterns



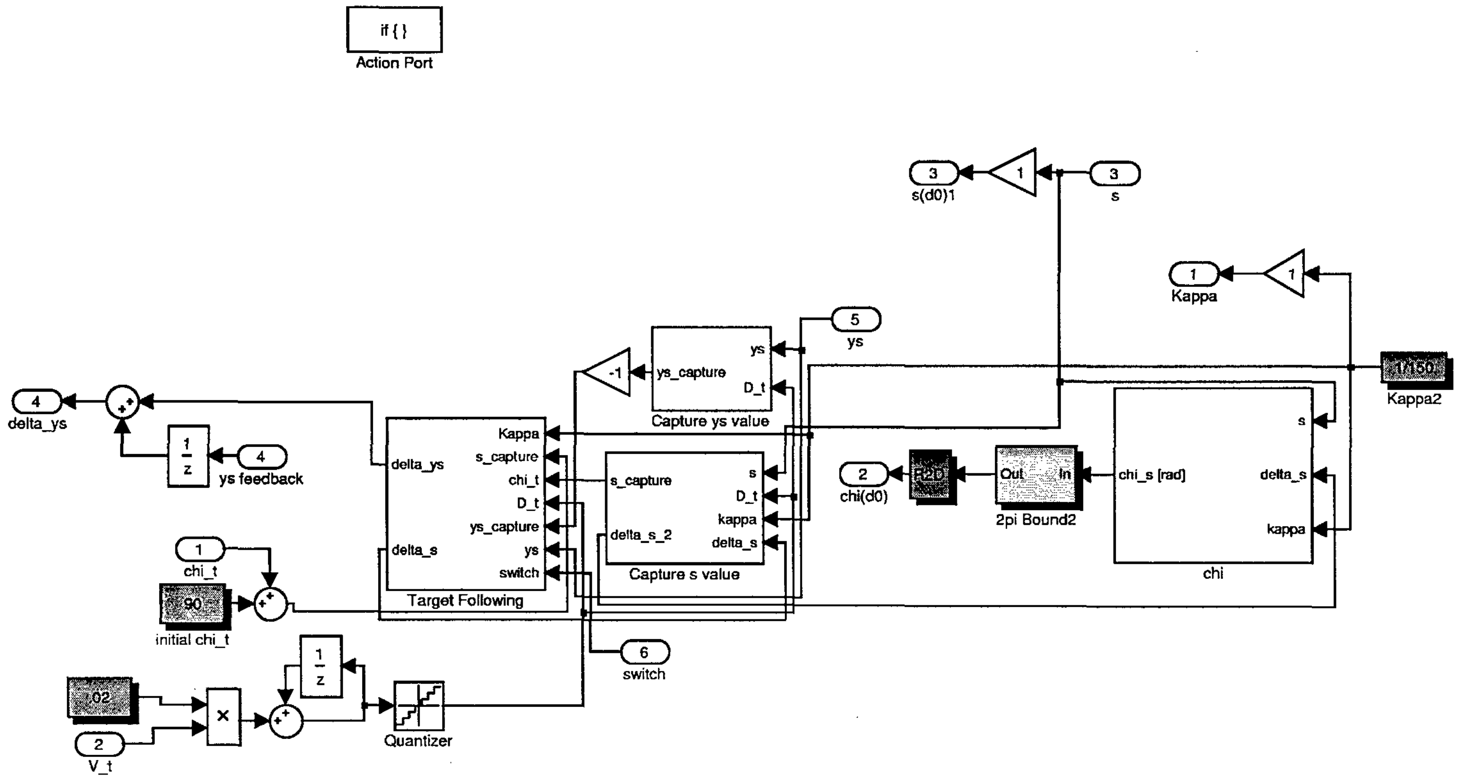
Square wave: Path following block



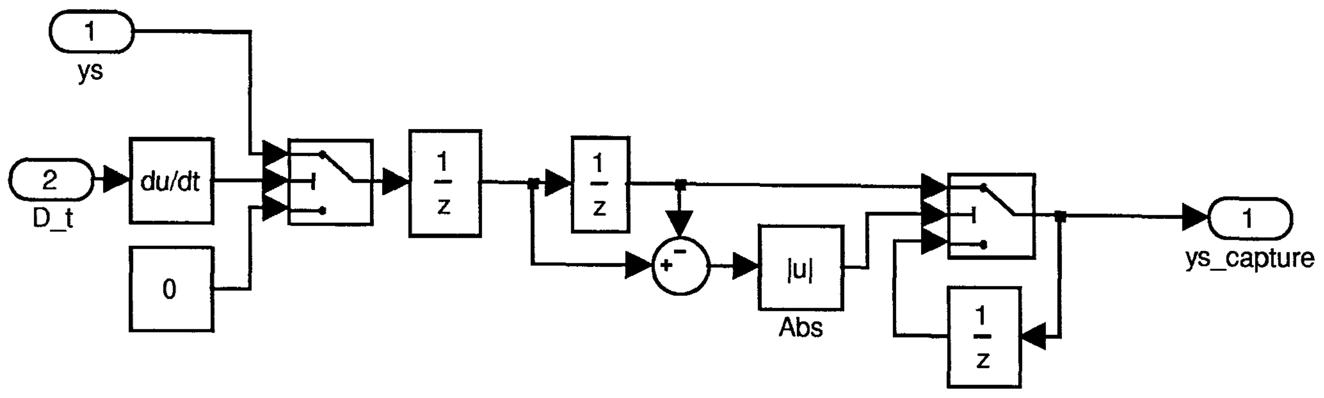
Square wave and Circle: Path definition block



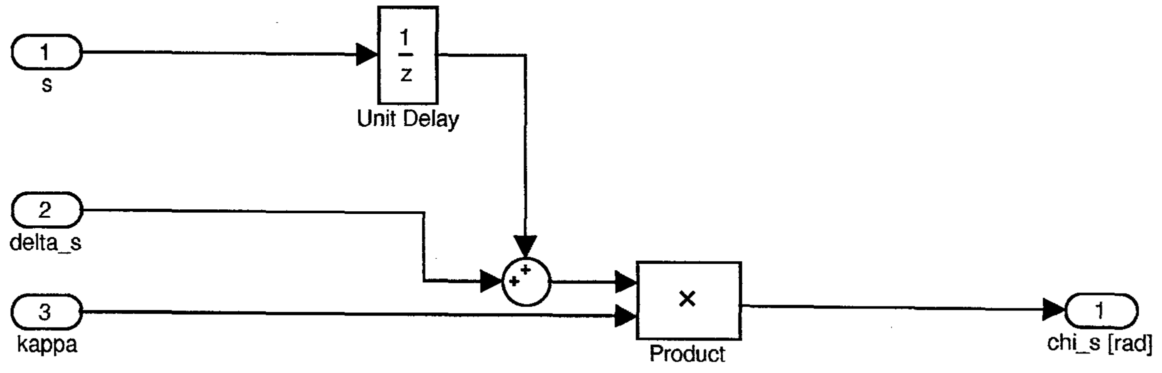
Square wave and Circle: Plot desired position block



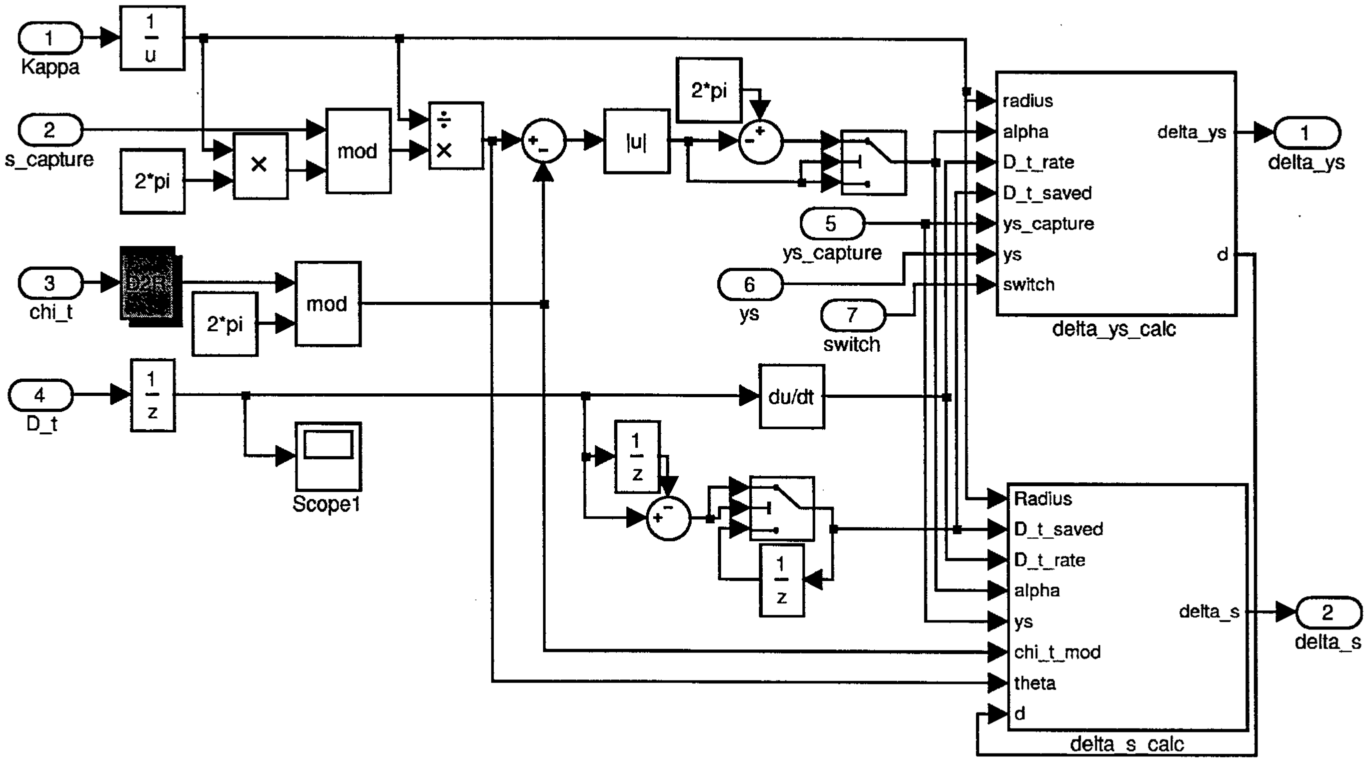
Square wave and Circle: Circular pattern block



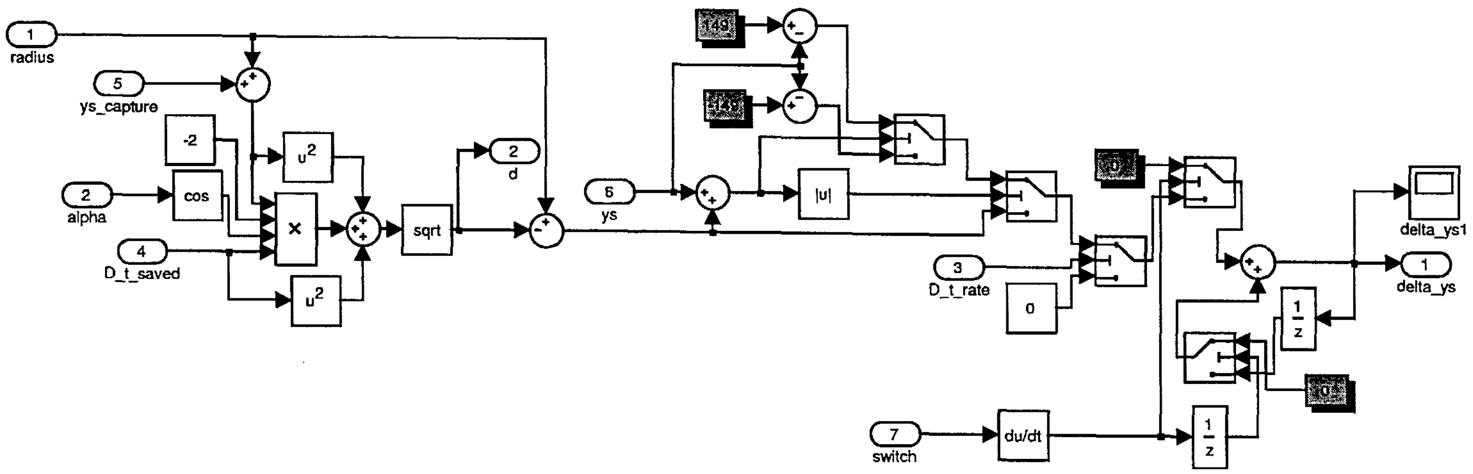
Square wave and Circle: Capture y_s value for circular path



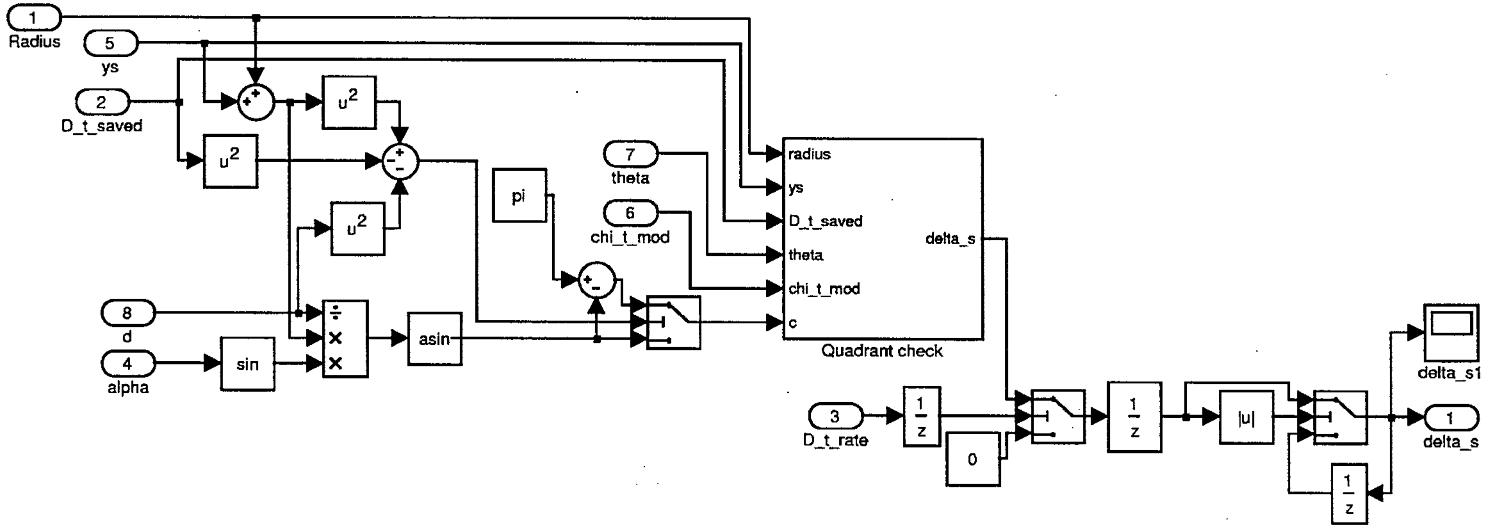
Square wave and Circle: Calculating χ for circular path



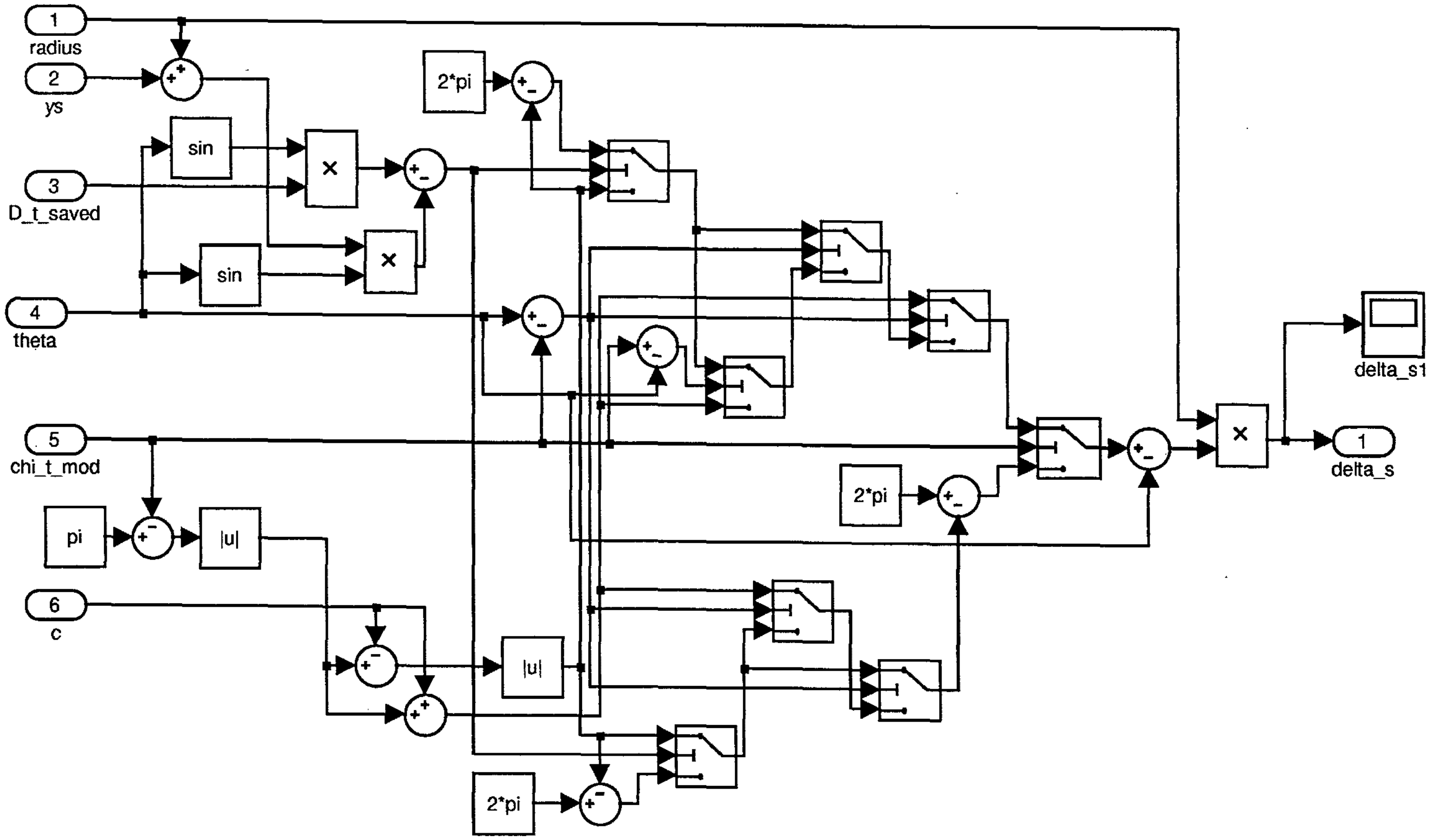
Square wave and Circle: Calculating Δ's for circular path



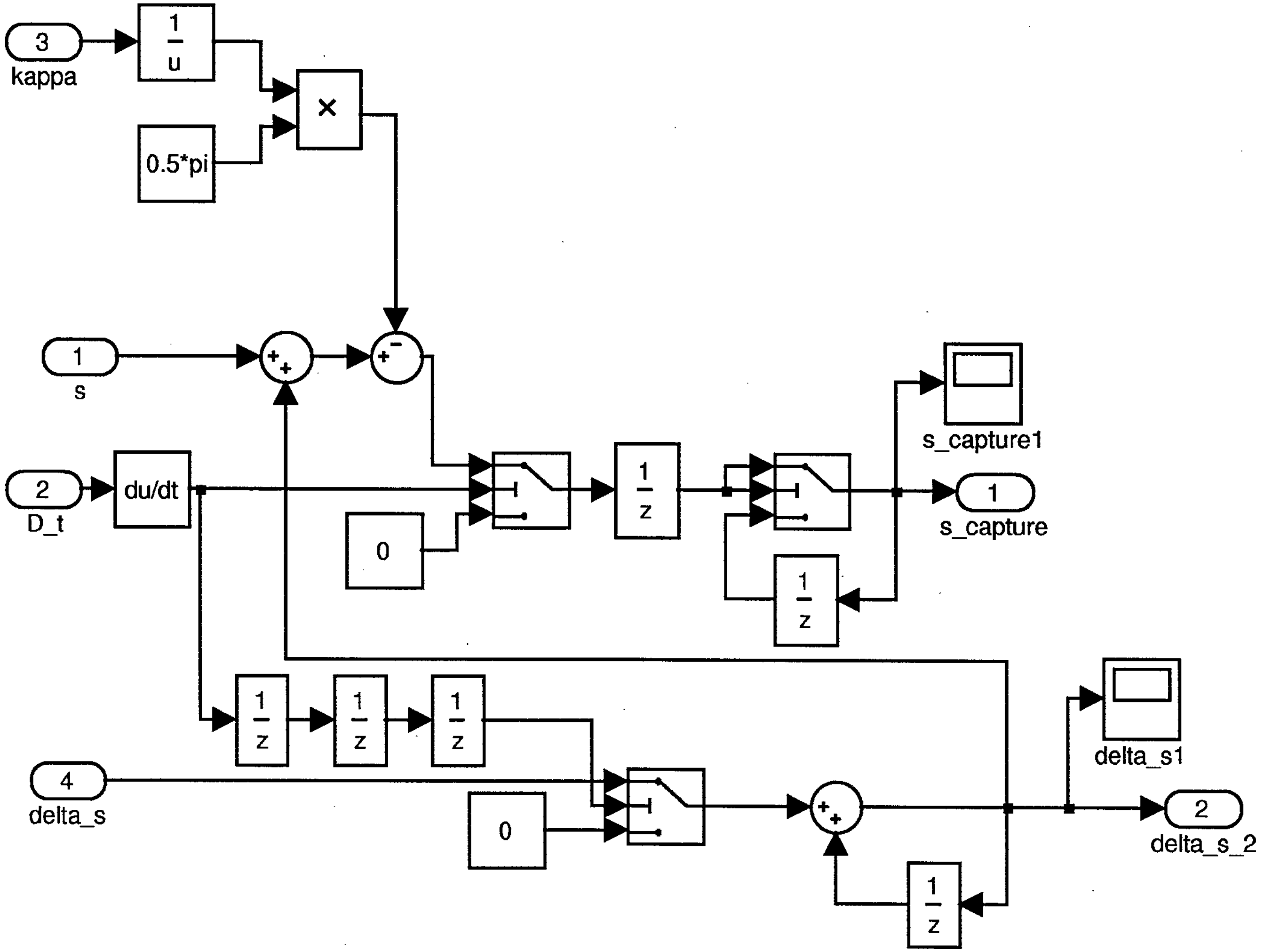
Square wave and Circle: Calculating Δy_s for circular path



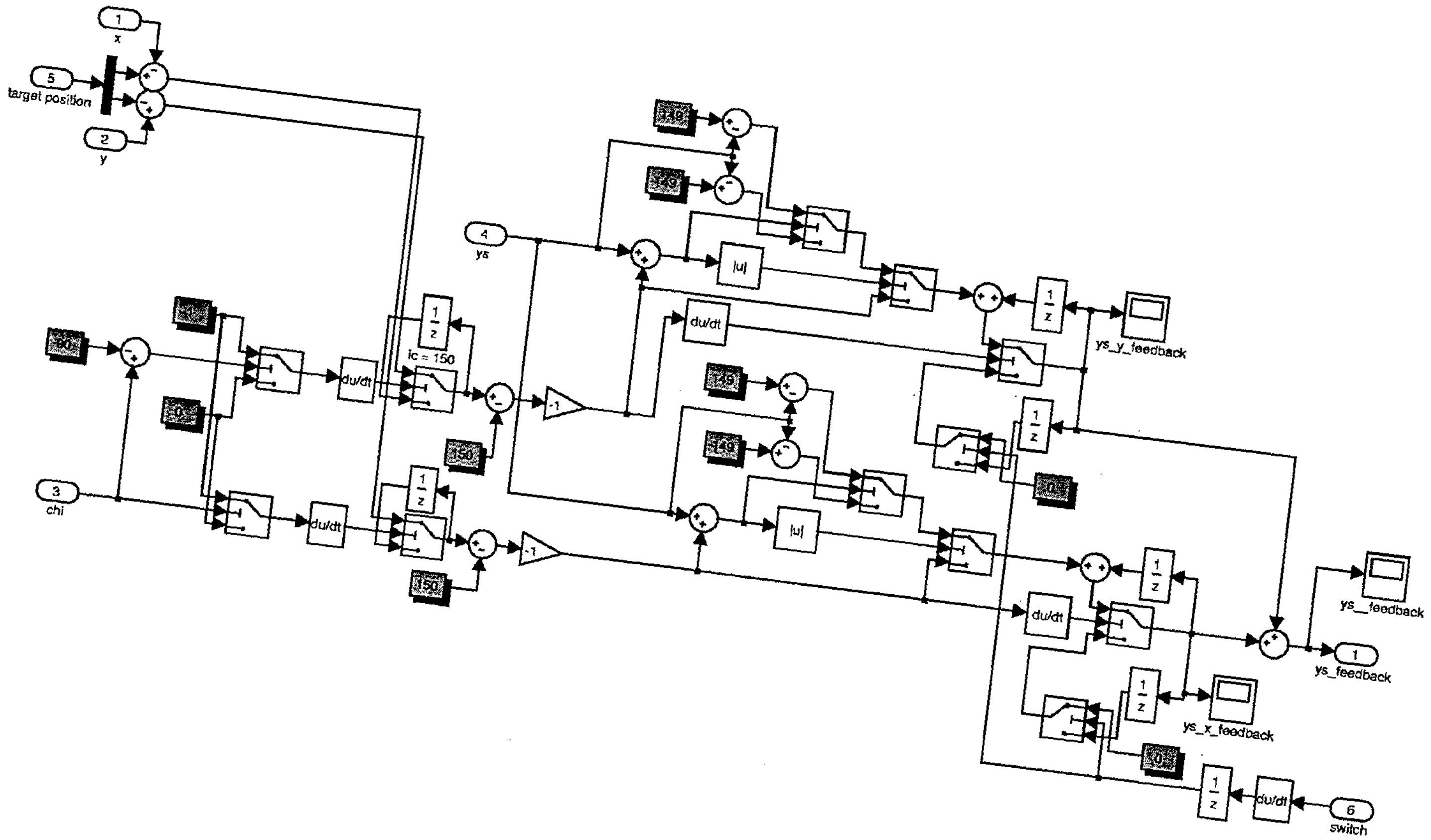
Square wave and Circle: Calculating Δs for circular path



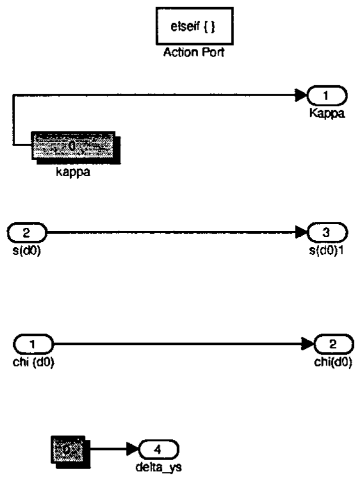
Square wave and Circle: Quadrant check for circular pattern



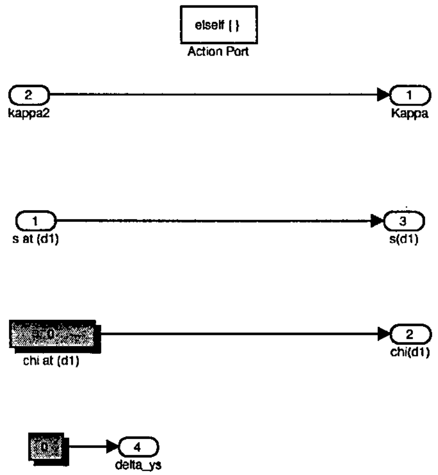
Square wave and Circle: Capture *s* value for circular path



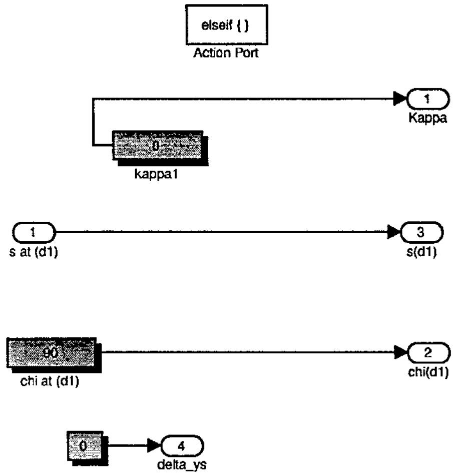
Square wave and Circle: Feedback block for the circular pattern



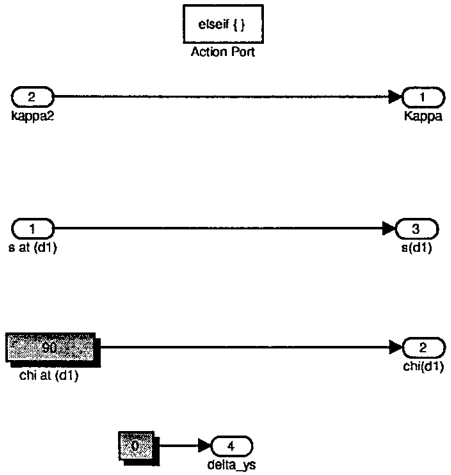
Square wave and Circle: Segment 1 of the square wave



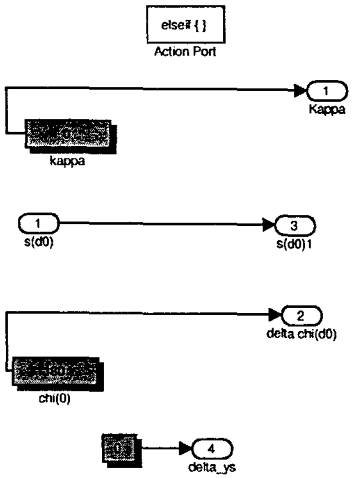
Square wave and Circle: Segment 2 of the square wave



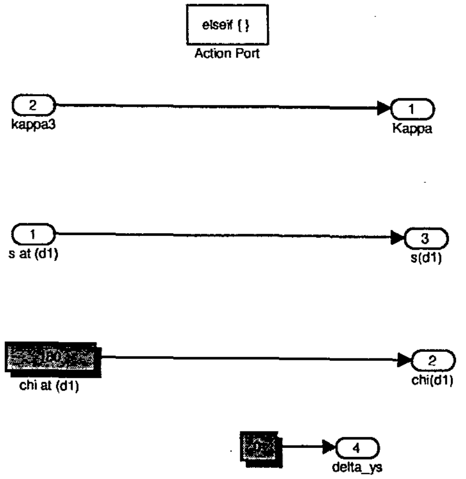
Square wave and Circle: Segment 3 of the square wave



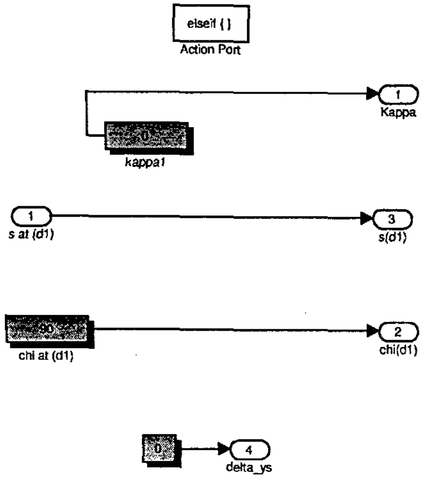
Square wave and Circle: Segment 4 of the square wave



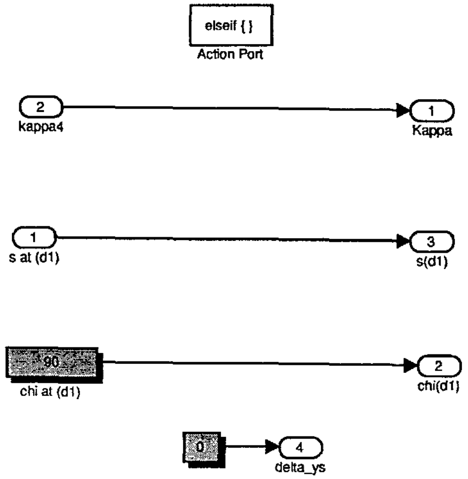
Square wave and Circle: Segment 5 of the square wave



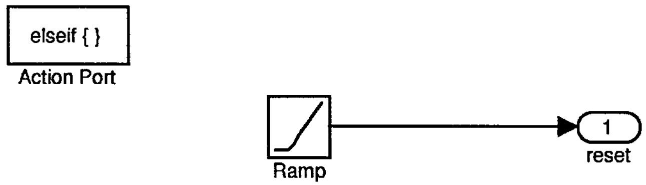
Square wave and Circle: Segment 6 of the square wave



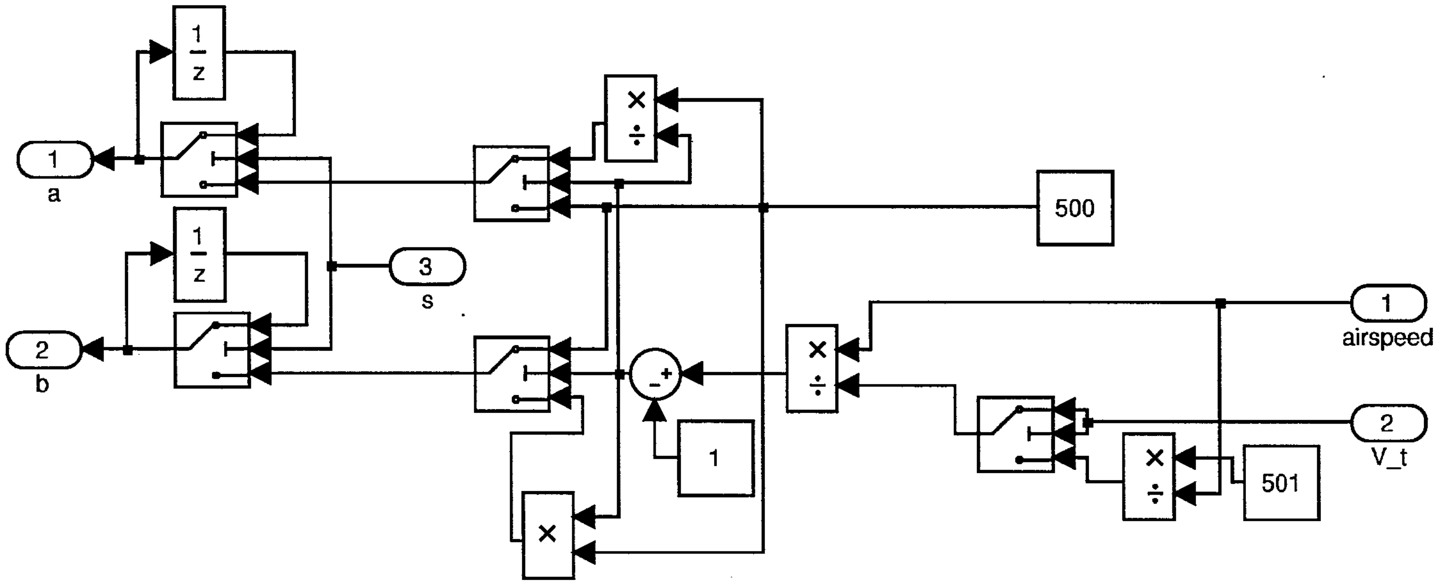
Square wave and Circle: Segment 7 of the square wave



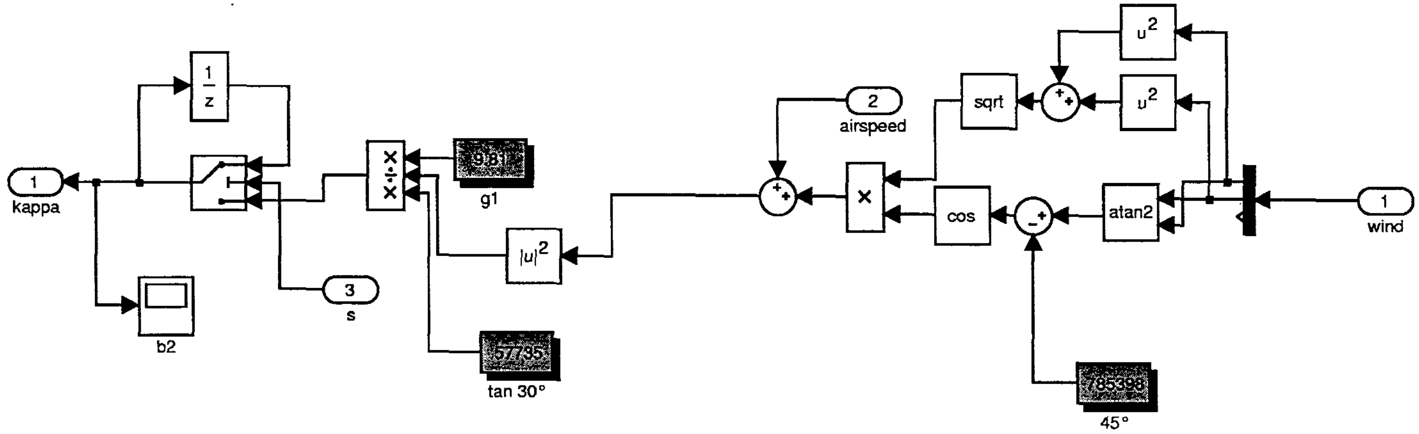
Square wave and Circle: Segment 8 of the square wave



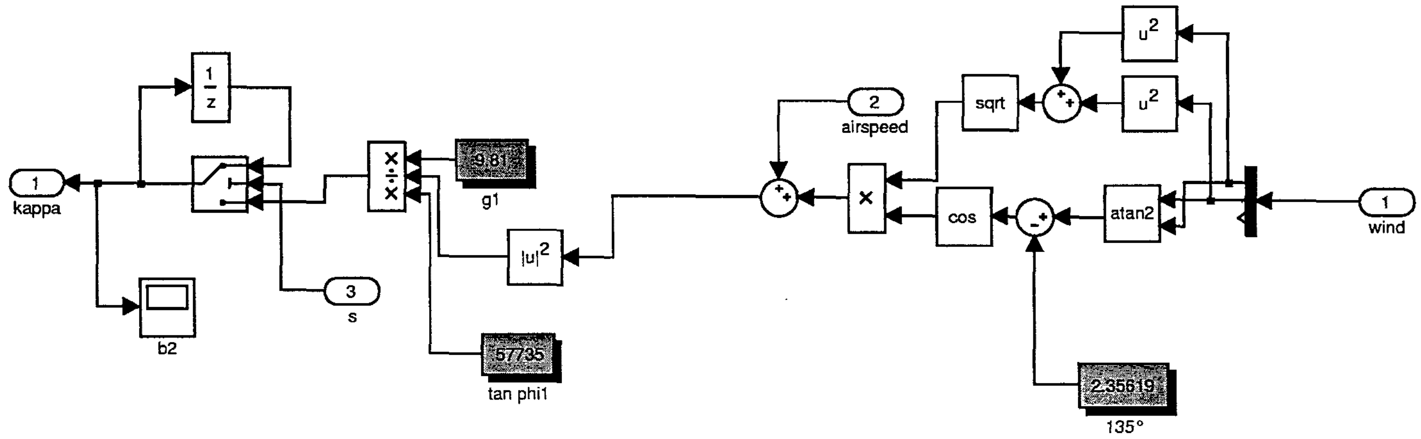
Square wave and Circle: Reset block



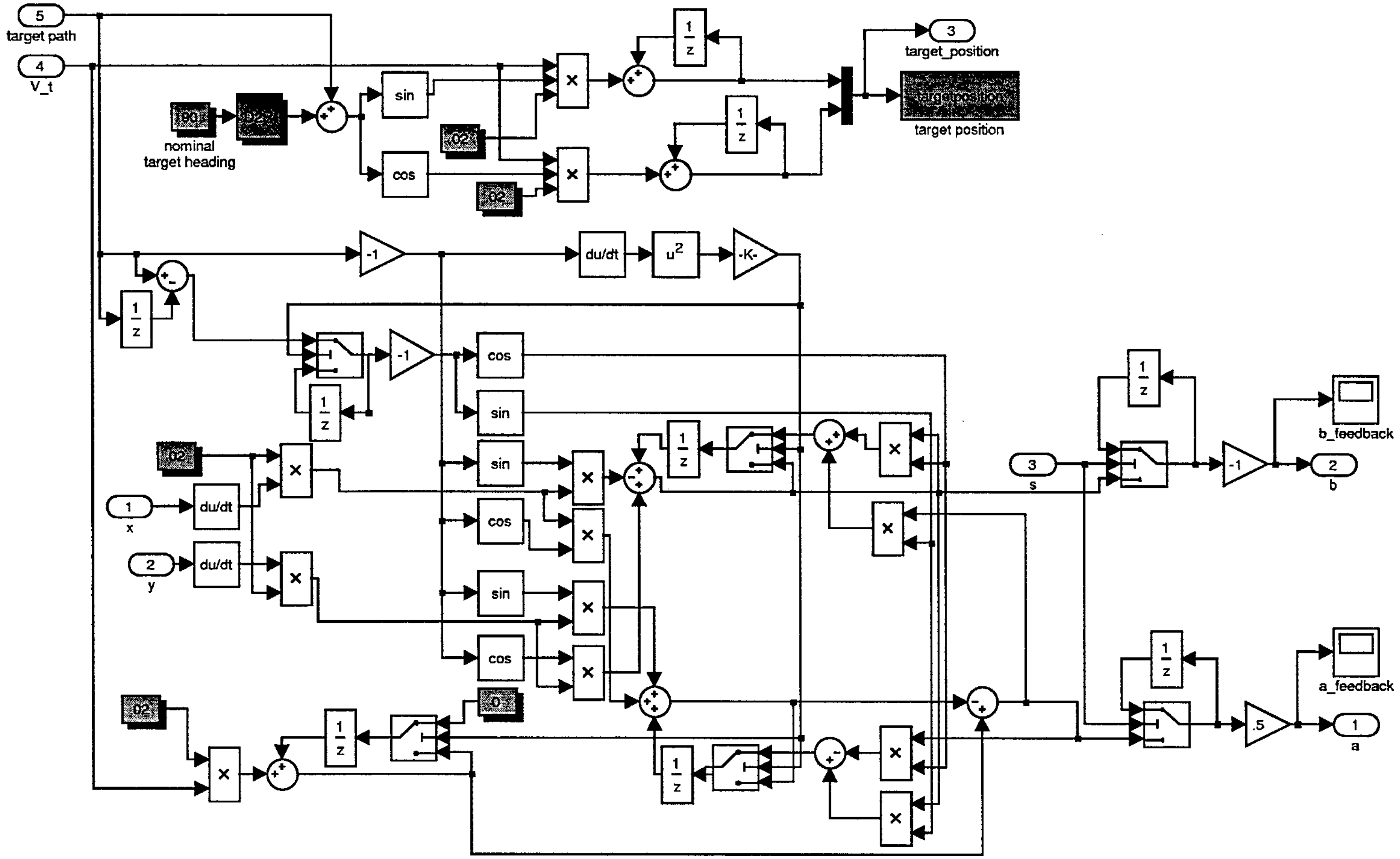
Square wave and Circle: Calculation of lengths a and b



Square wave and Circle: Calculations for corners 1 and 4 of the square wave pattern

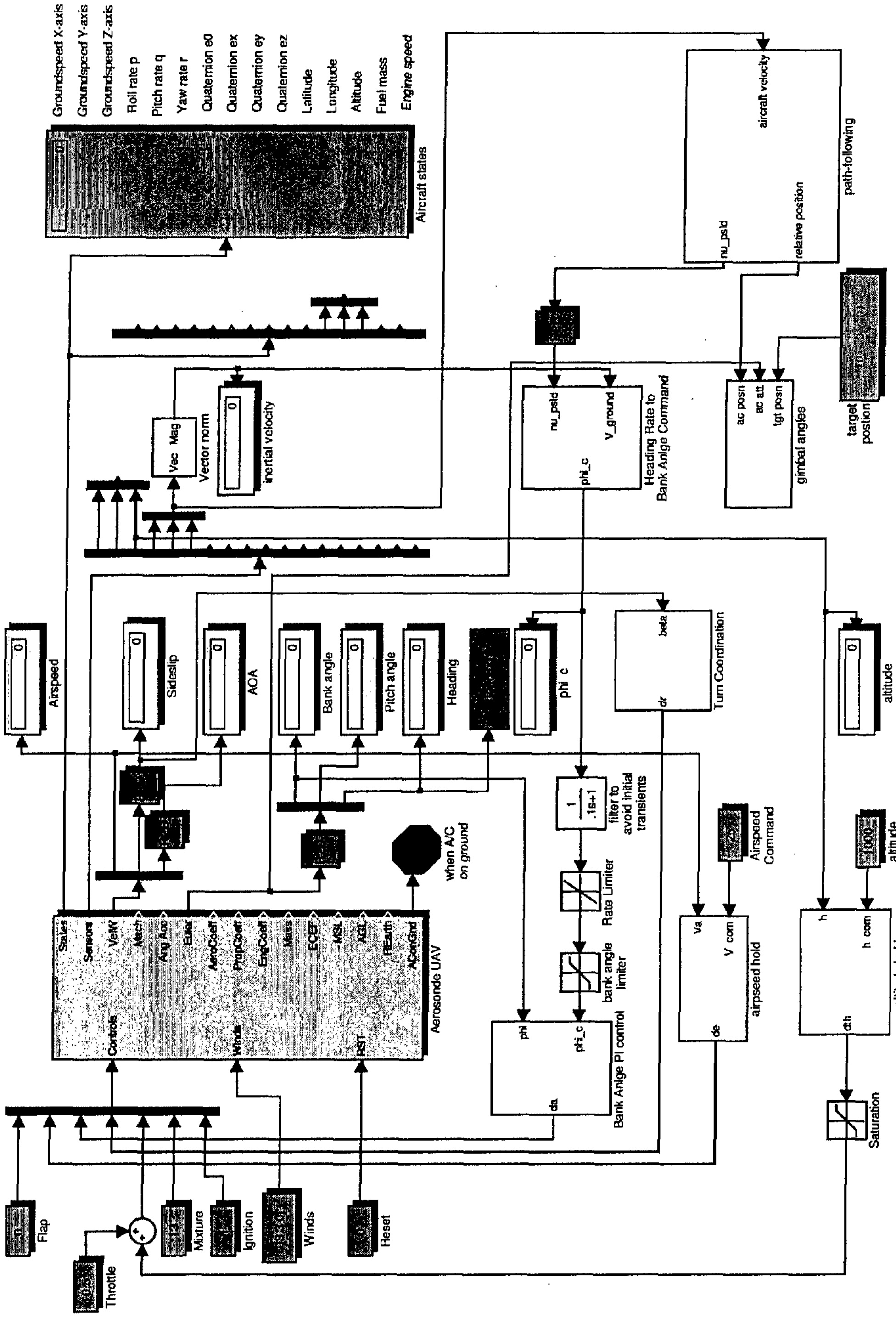


Square wave and Circle: Calculation for corners 2 and 3 of the square wave pattern

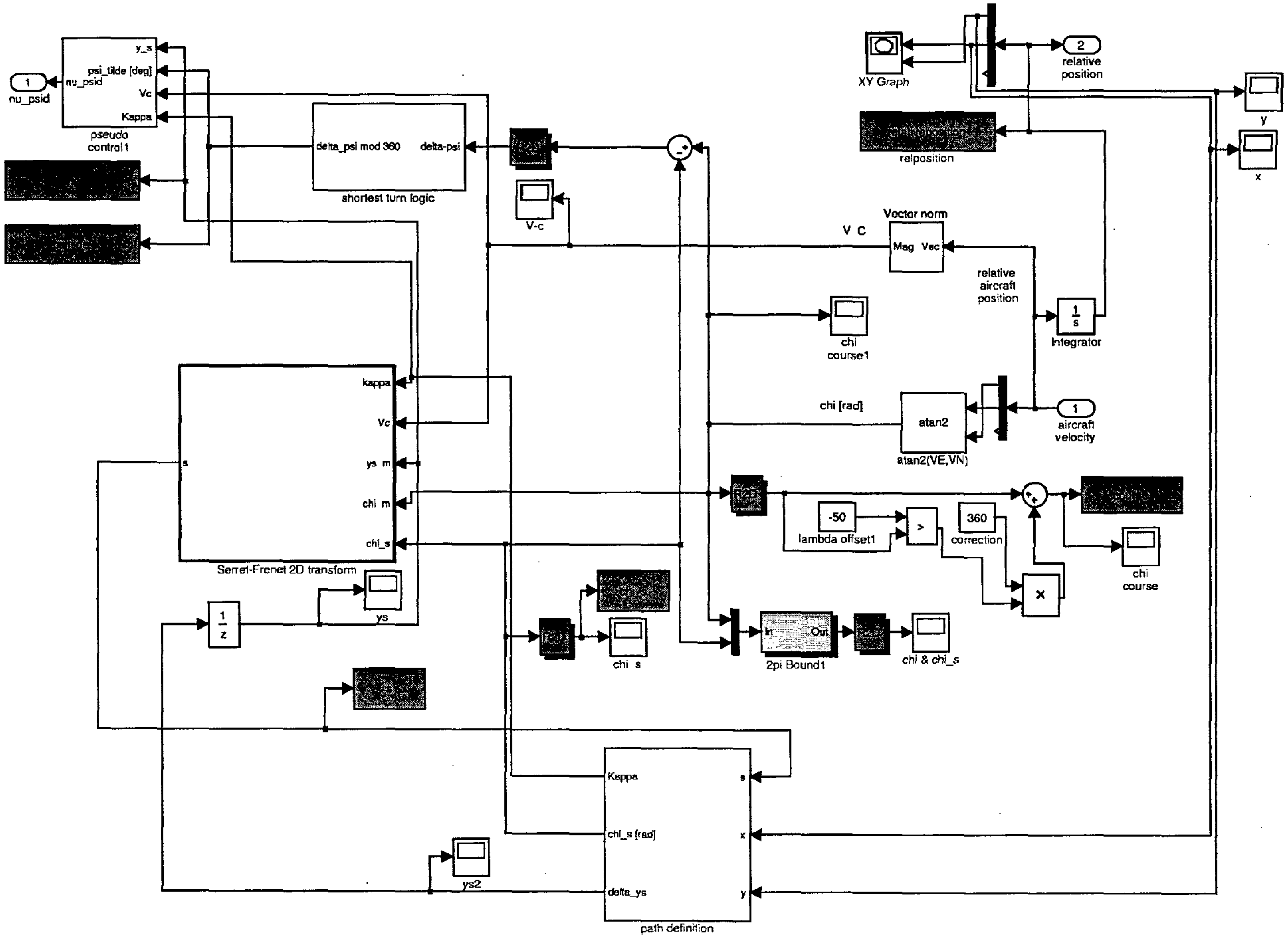


Square wave and Circle: Feedback block for the square wave pattern

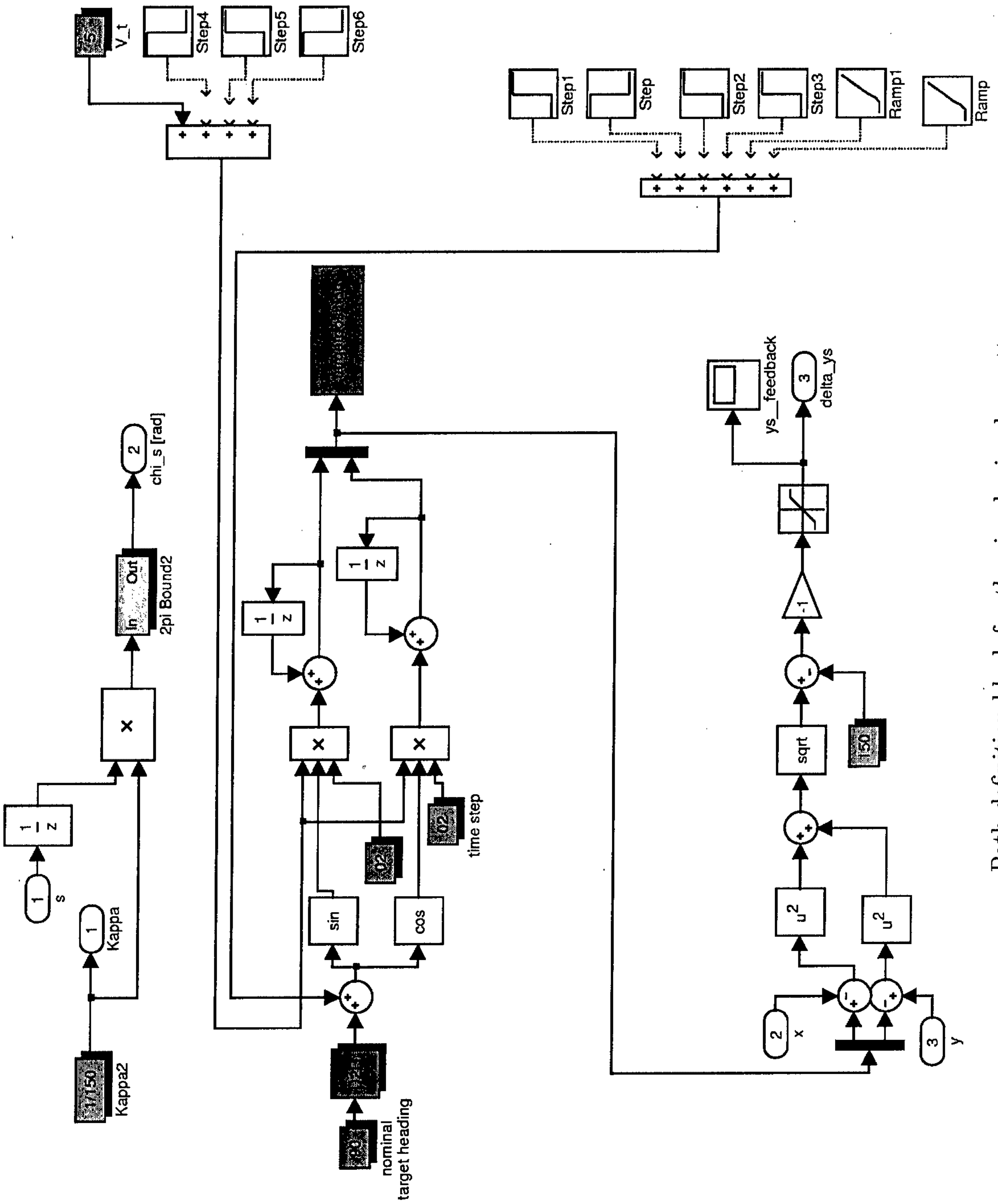
B.3 Simulink Model for the Simple Circular Pattern



Main Aerosonde Simulink model for the simple circular pattern

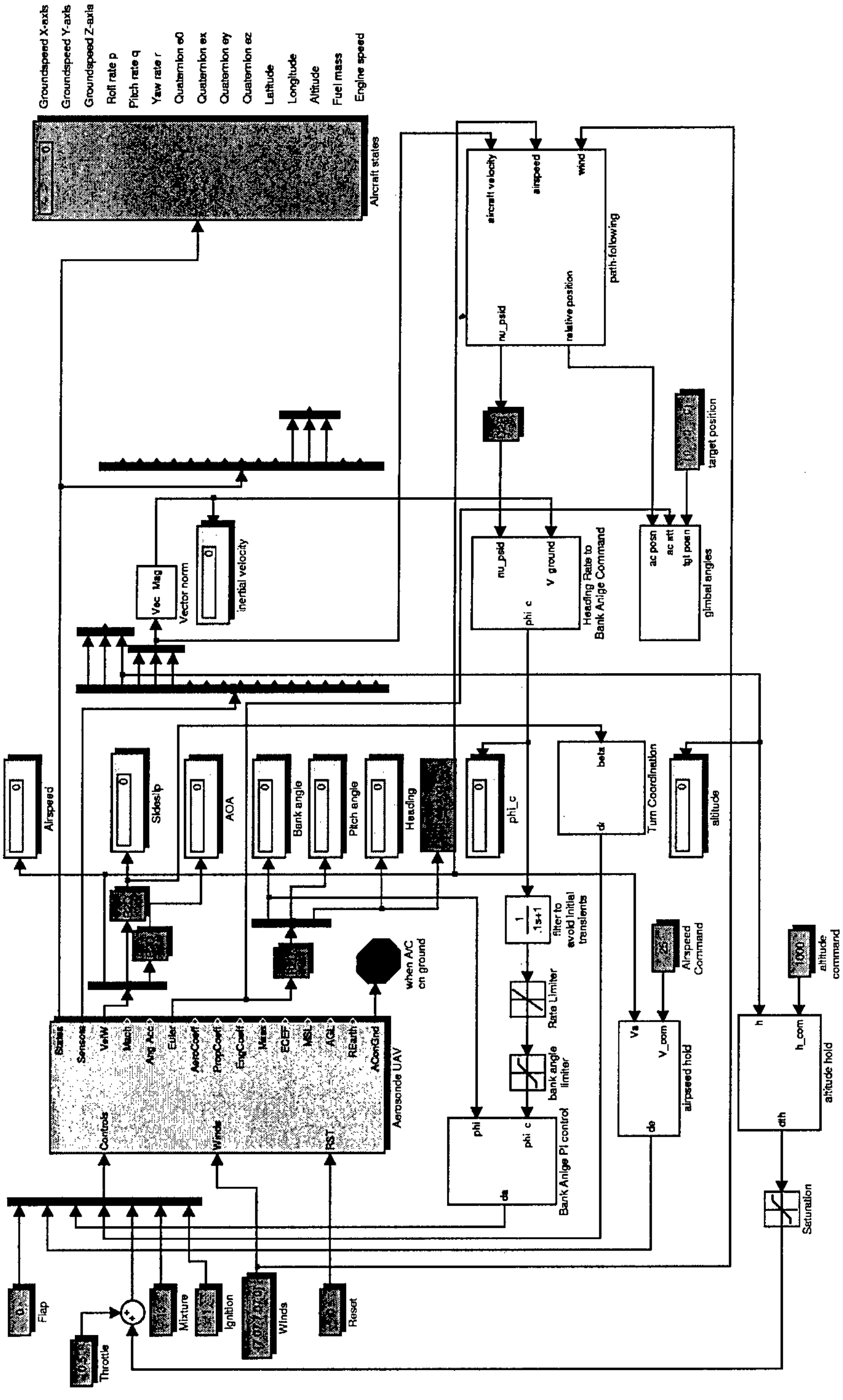


Path following block for the simple circular pattern

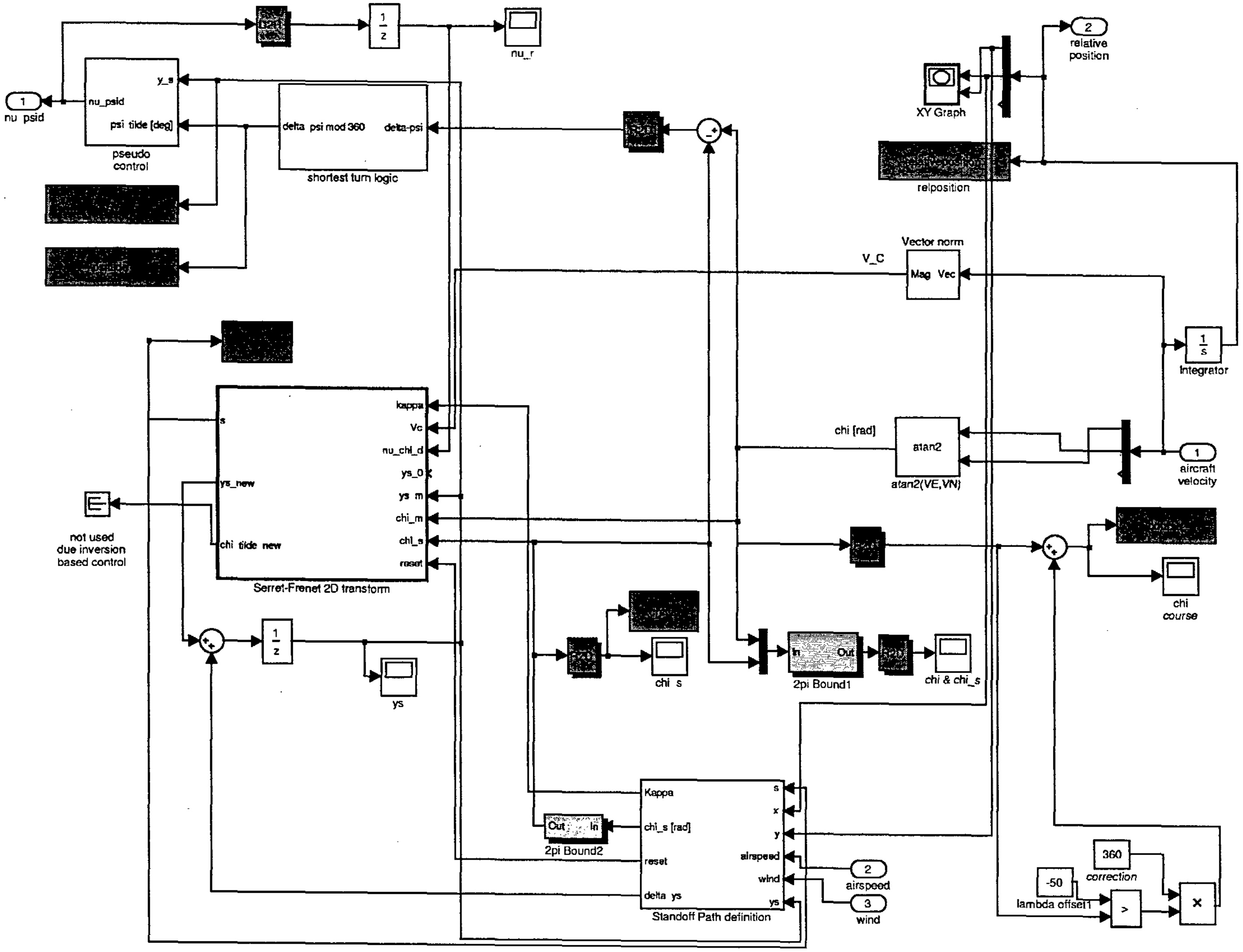


Path definition block for the simple circular pattern

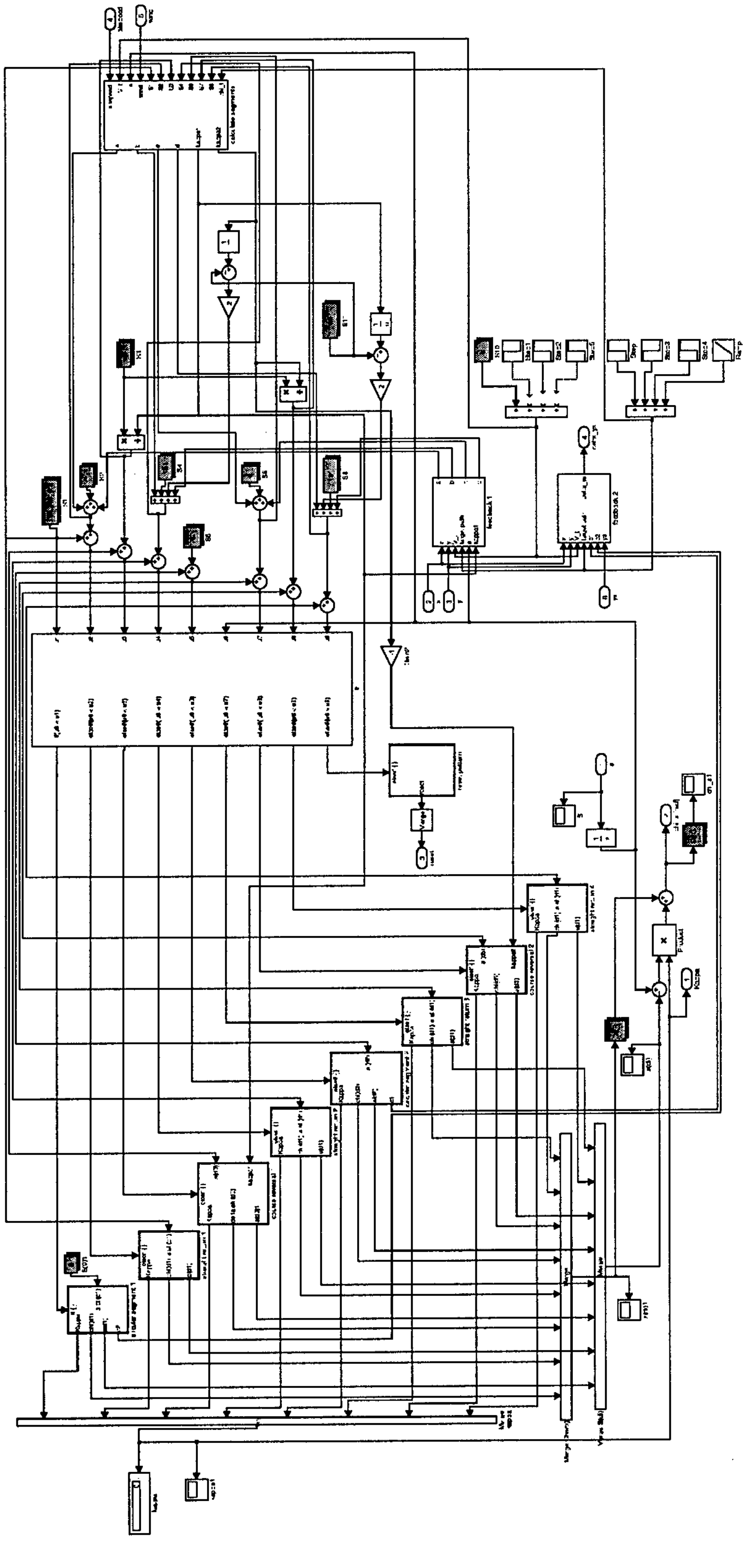
B.4 Simulink Model for the Standoff Pattern



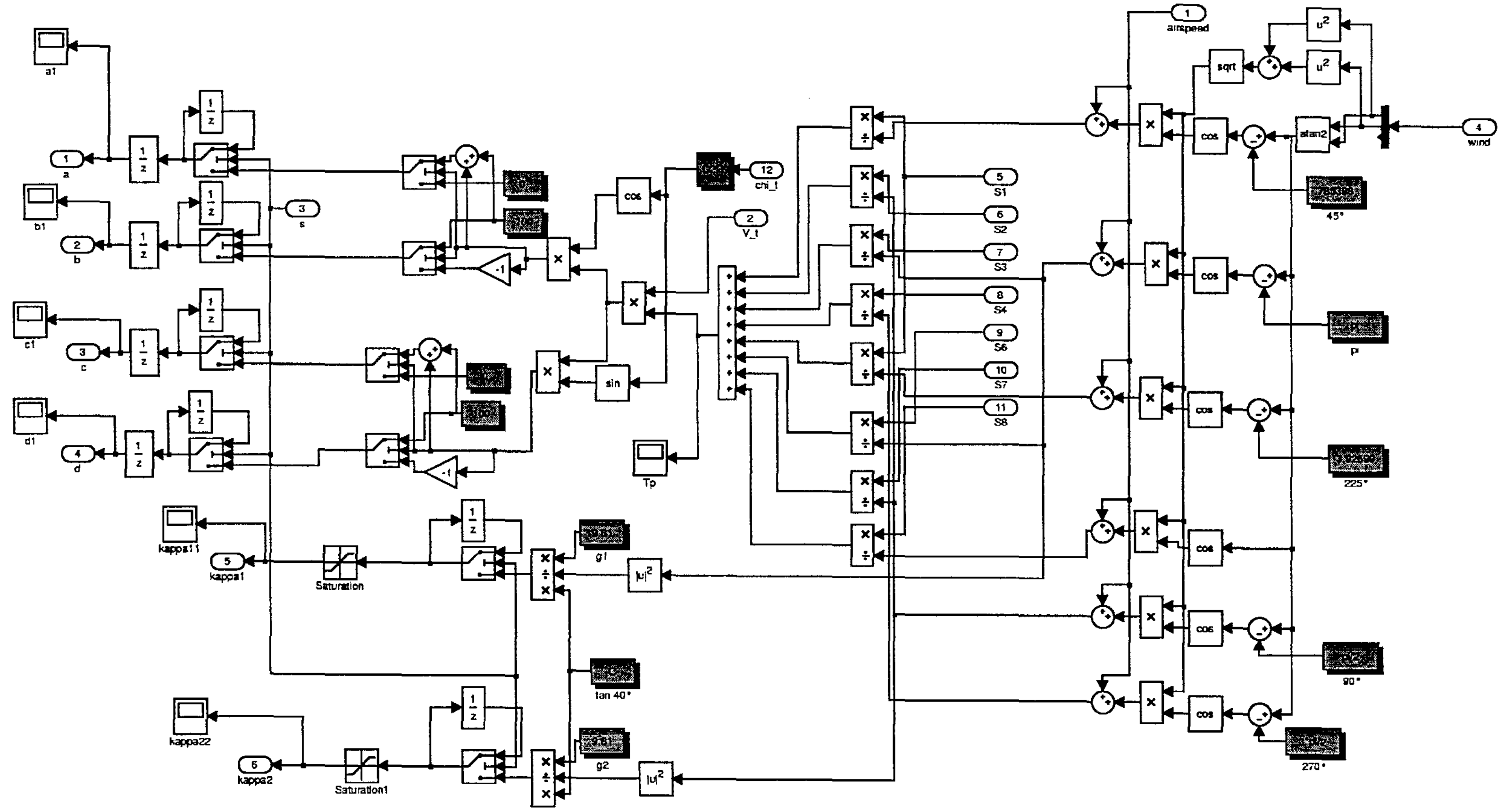
Main Aerosonde Simulink model for the standoff pattern



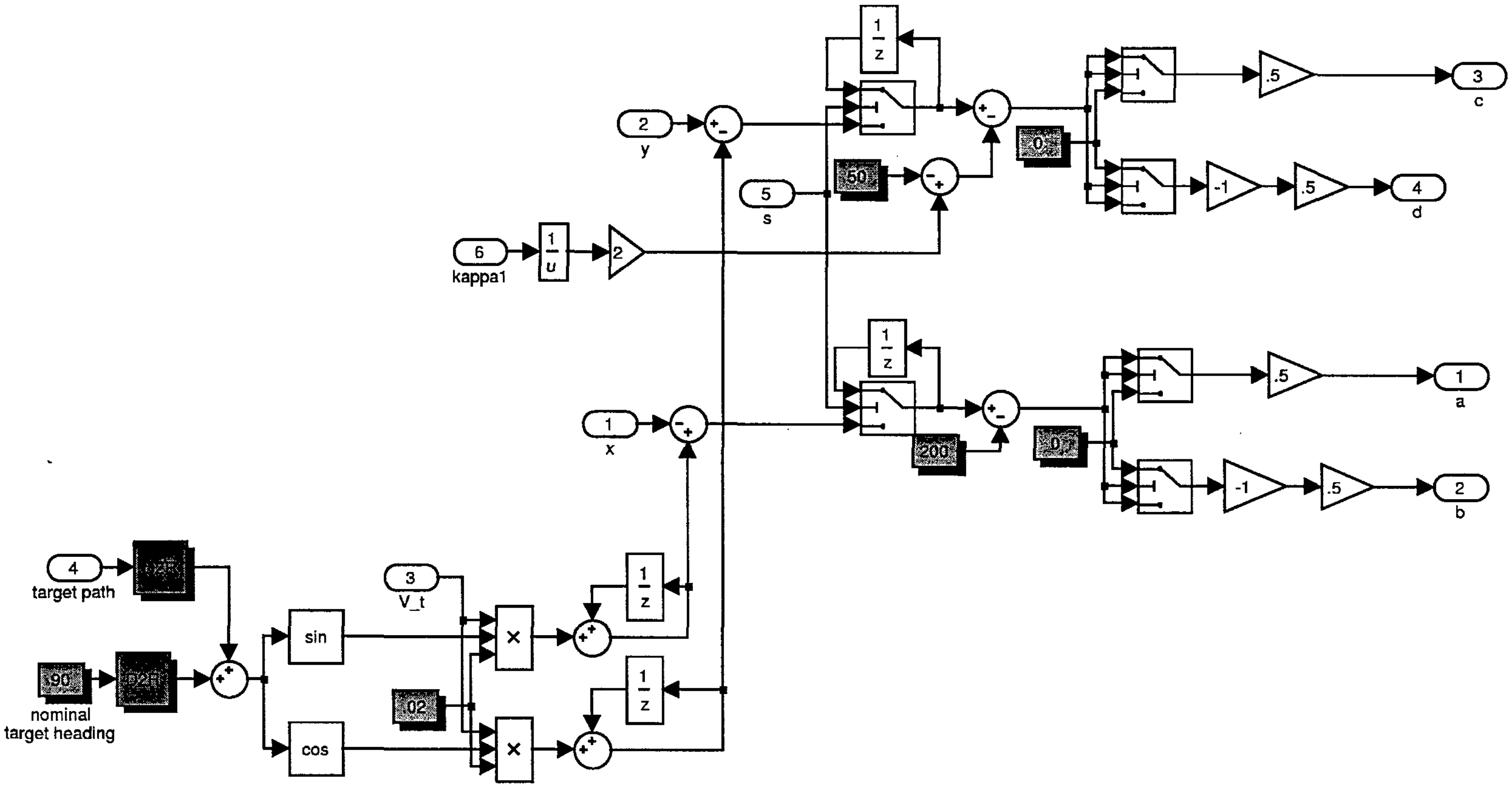
Path following block for the standoff pattern



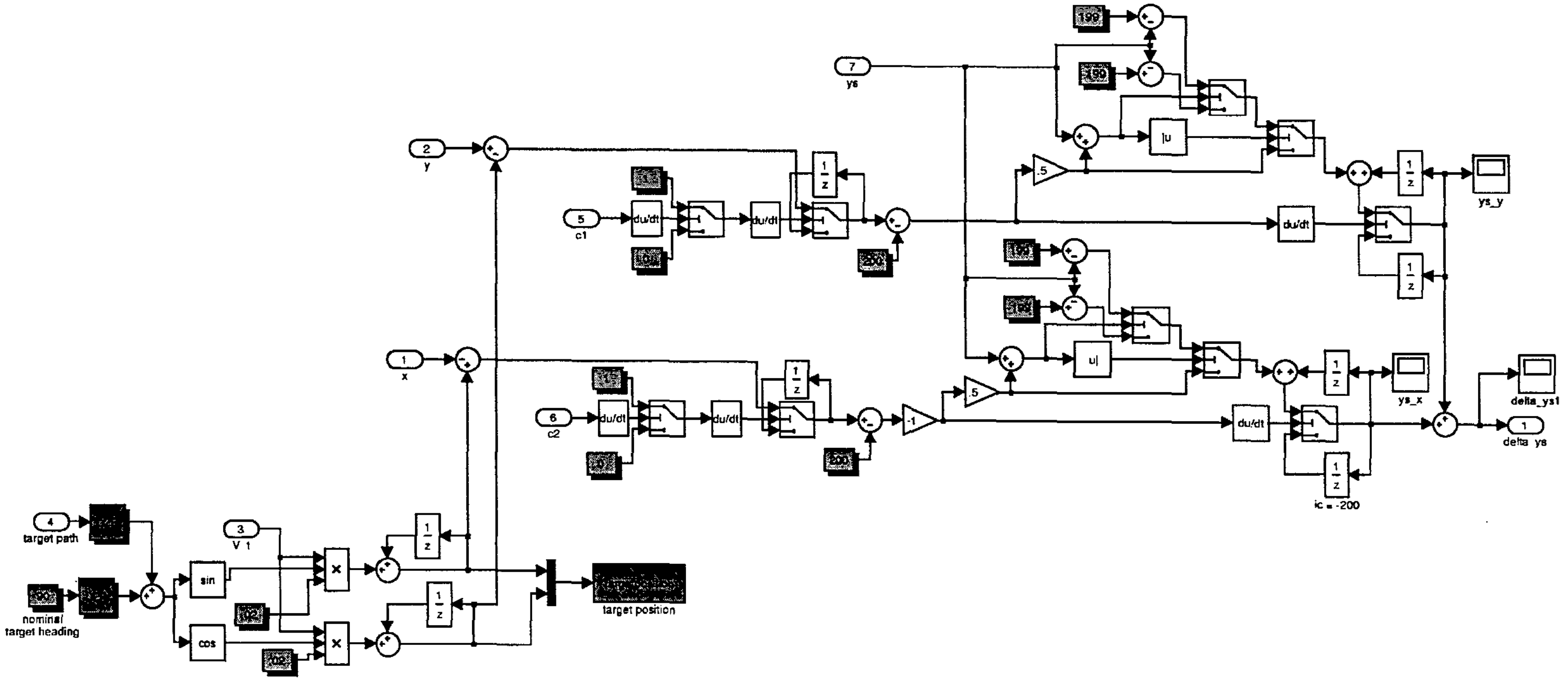
Path definition block for the standoff pattern



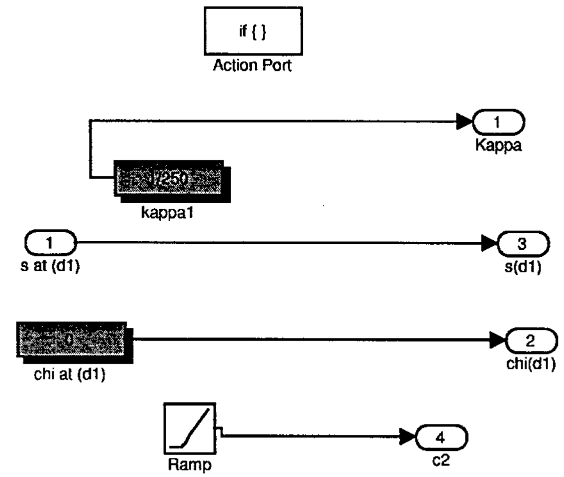
Calculations for segment lengths for the standoff pattern



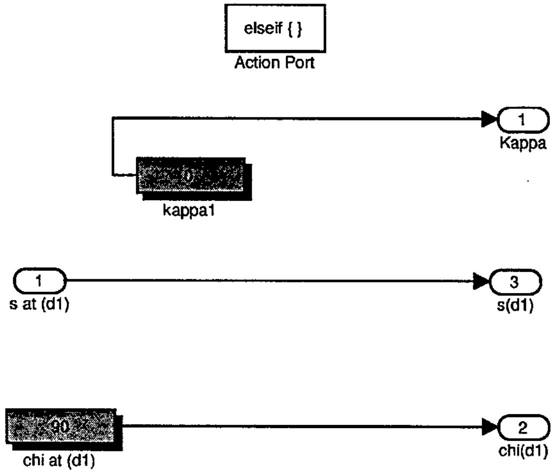
Abcd feedback block for the standoff pattern



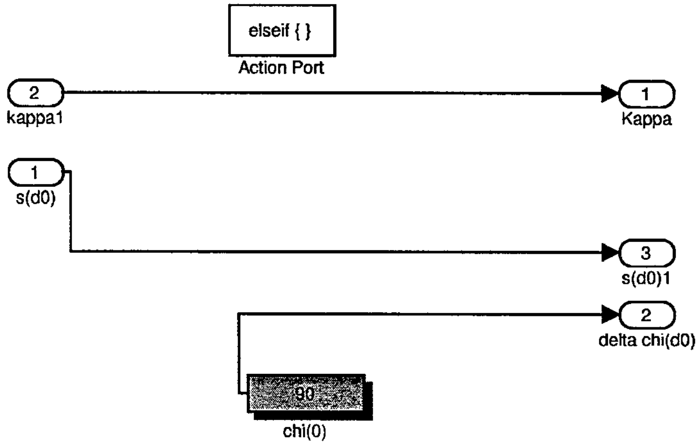
y_s feedback block for the standoff pattern



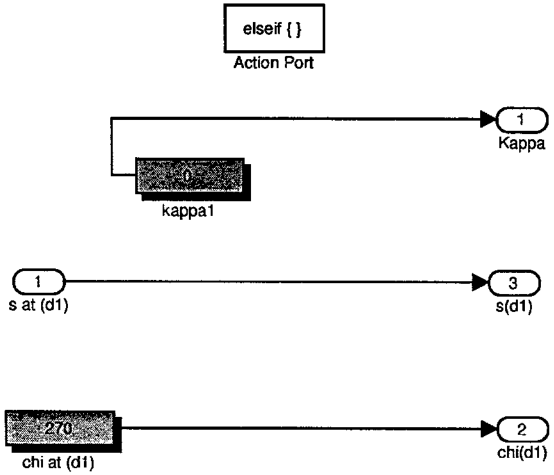
Segment 1 of standoff pattern



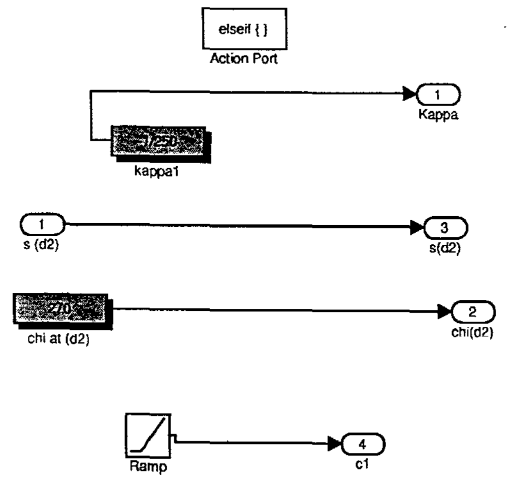
Segment 2 of standoff pattern



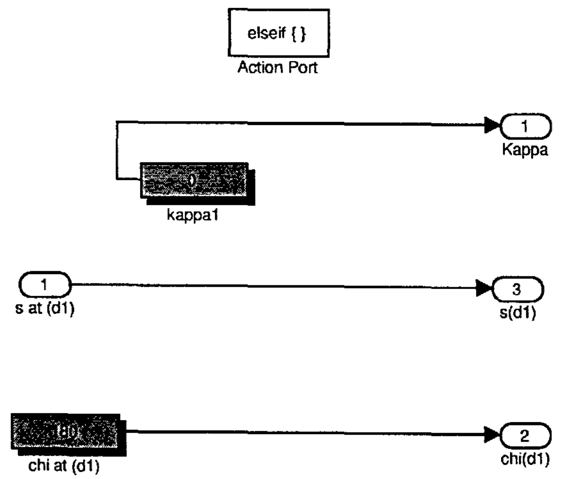
Segment 3 of standoff pattern



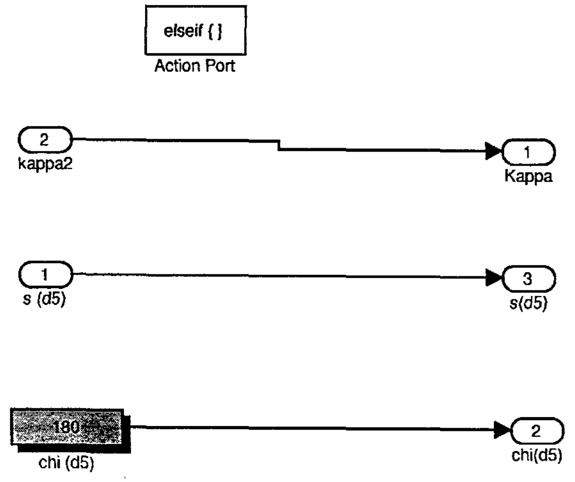
Segment 4 of standoff pattern



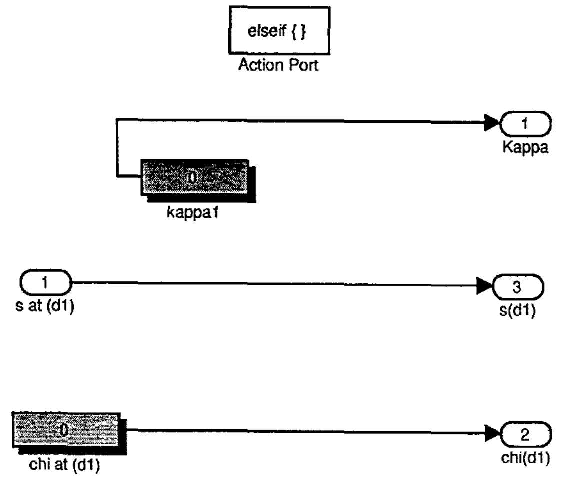
Segment 5 of standoff pattern



Segment 6 of standoff pattern



Segment 7 of standoff pattern



Segment 8 of standoff pattern